

**PEMBANGUNAN SISTEM APLIKASI REKAM MEDIS
ELEKTRONIK DAN PENDAFTARAN PASIEN ONLINE
BERBASIS WEB (STUDI KASUS : KLINIK MEDIS ELISA
MALANG)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Clara Pusparani
NIM: 145150207111078



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PEMBANGUNAN SISTEM APLIKASI REKAM MEDIS ELEKTRONIK DAN
PENDAFTARAN PASIEN ONLINE BERBASIS WEB (STUDI KASUS : KLINIK MEDIS
ELISA MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Clara Pusparani
NIM: 145150207111078

Skripsi ini telah diuji dan dinyatakan lulus pada
7 November 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Bayu Priyambadha, S.Kom, M.Kom
NIP. 19820909 200812 1 004

Dosen Pembimbing II



Achmad Arwan, S.Kom, M.Kom
NIP . 19840815 200812 1 004

Mengetahui
Ketua Jurusan Teknik Informatika



Astoro Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 November 2018



Clara Pusparani

NIM: 145150207111078

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yesus Kristus atas segala berkat dan penyertaan-Nya selama penelitian ini sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pembangunan Sistem Aplikasi Rekam Medis Elektronik Dan Pendaftaran Pasien *Online* Berbasis Web (Studi Kasus : Klinik Medis Elisa Malang)”.

Dalam proses penyusunan skripsi ini, penulis memperoleh bantuan dan bimbingan dari banyak pihak baik secara moril maupun materil. Oleh karena itu, pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Bayu Priyambadha, S.Kom, M.Kom, selaku dosen pembimbing I atas waktu, tenaga, dan ilmu yang diberikan untuk membimbing penulis selama proses penyusunan skripsi ini.
2. Bapak Achmad Arwan, S.Kom, M.Kom, selaku dosen pembimbing II atas waktu, tenaga, dan ilmu yang diberikan untuk membimbing penulis selama proses penyusunan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T, Ph.D., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Drs. Mardji, M.T, dan Bapak Edy Santoso, S.Si., M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Seluruh dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas waktu, ilmu dan pengetahuan yang diberikan kepada penulis.
6. Ibu drg. Elly Nawangsih, Bapak Renaldo, dan seluruh staf Klinik Medis Elisa, yang telah memberikan informasi yang dibutuhkan penulis selama proses penelitian.
7. Kedua orang tua penulis, Bapak Yohanes S. Pudjiarwanto dan Ibu Elly Nawangsih, juga kepada adik-adik penulis, Sonya Maharani, Isabela Anjani, dan seluruh keluarga penulis atas doa, dukungan, dan materi yang diberikan kepada penulis.
8. Bonang Ananta Sakti, yang selalu memberikan dukungan dan semangat kepada penulis.
9. Sahabat-sahabat penulis, Grace Angela dan Monica Dian atas dukungan yang tak pernah henti kepada penulis.
10. Teman-teman penulis selama mengerjakan penelitian ini, Anton Firdaus, Ditya Enandini, Tafarrara Irsa, Fendra Gunawan, Yudha Yusi, Andriko Hedi, Nur Afdaliyah, Sastra Ginata, Nurhaida Syadrina, Yoga Rizwan, dan Shinta Aprilisia yang telah meluangkan waktu untuk menyusun skripsi bersama dan berbagi ilmu.

11. Teman-teman KOPMA SQUAD, yang berproses bersama selama perkuliahan dan selalu memberikan semangat dalam proses pengerjaan skripsi.
12. Teman-teman PSM Brawijaya, atas kebersamaan dalam berbagai kegiatan kepanitiaan, lomba, dan kegiatan lain sehingga menjadi sarana bagi penulis untuk mengembangkan talenta.
13. Seluruh pihak yang membantu penulis selama pengerjaan skripsi, yang tidak bisa disebutkan satu per satu di sini. Terima kasih atas dukungan dan doa yang telah diberikan.

Malang, 7 November 2018

Penulis

clarapusparani@student.ub.ac.id



ABSTRAK

Clara Pusparani, Pembangunan Sistem Aplikasi Rekam Medis Elektronik dan Pendaftaran Pasien *Online* Berbasis Web (Studi Kasus : Klinik Medis Elisa Malang)

Pembimbing: Bayu Priyambadha, S.Kom, M.Kom dan Achmad Arwan, S.Kom, M.Kom.

Manajemen pengelolaan informasi adalah hal utama yang harus dipenuhi oleh perusahaan khususnya bila memiliki data besar dan kompleks. Saat ini telah banyak rumah sakit, klinik, puskesmas, maupun praktik dokter yang telah menggunakan sistem informasi untuk memudahkan manajemen data dan meningkatkan mutu pelayanan kepada pasien. Namun Klinik Medis Elisa Malang masih menemui kendala dalam pendaftaran pasien dan pencarian rekam medis. Satu orang pasien bisa memakan waktu 5-15 menit untuk melakukan proses pendaftaran, termasuk pencarian rekam medis pasien atau pembuatan dokumen rekam medis untuk pasien yang pertama kali berobat. Proses ini dapat memakan waktu lebih lama lagi jika dokumen rekam medis pasien ternyata tidak ditemukan atau ditemukan dalam keadaan rusak, petugas harus membuat rekam medis yang baru. Melihat permasalahan tersebut, penulis mengembangkan sistem rekam medis elektronik dan pendaftaran pasien *online* berbasis web yang menyediakan fitur pendaftaran berobat untuk pasien lama dan juga memiliki fitur-fitur yang berkaitan dengan manajemen rekam medis pasien yang sudah dikonversi menjadi bentuk elektronik. Penulis menggunakan *software development life cycle* dengan model *waterfall*. Pengujian dilakukan menggunakan pengujian unit dengan metode *whitebox testing* dan pengujian validasi dengan metode *blackbox testing* yang menghasilkan nilai *valid* pada semua kasus uji, serta pengujian *compatibility* dengan hasil 82 % valid.

Kata kunci: rekam medis elektronik, rekayasa perangkat lunak, *waterfall model*

ABSTRACT

Clara Pusparani, Pembangunan Sistem Aplikasi Rekam Medis Elektronik dan Pendaftaran Pasien Online Berbasis Web (Studi Kasus : Klinik Medis Elisa Malang)

Adviser: Bayu Priyambadha, S.Kom, M.Kom and Achmad Arwan, S.Kom, M.Kom.

Management information is the main thing that has to be met by companies with a large and complex data. Currently, many hospitals, clinics, health centers, as well as the practices of doctors have used information system to facilitate data management and improve the quality of service to patients. But Klinik Elisa Malang encountered obstacles in registration of patients and medical record search. One patient could take 5-15 minutes to perform the registration process, including the search of the patient's medical record or creating medical record documents for new patients. This process can take much longer if a patient's medical record document was not found or is found in the broken state, the officer should make a new medical record. Based on these problems, the authors developed the electronic medical record system and online patient registration web-based that provides medical registration for old patients and also has features that are related to the patient's medical record management that have been converted to electronic form. The author uses the software development life cycle with a waterfall model. Testing is done using unit testing with whitebox method and validation testing using blackbox method that produces valid values on all test cases, and compatibility test with 82% valid results.

Keywords: *electronic medical record, software engineering, waterfall model*

DAFTAR ISI

KATA PENGANTAR.....	iii
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Rekam Medis.....	6
2.2.1 Rekam Medis Elektronik.....	7
2.3 <i>System Development Life Cycle</i>	9
2.3.1 <i>Model Waterfall</i>	9
2.4 UML (<i>Unified Modeling Language</i>).....	10
2.4.1 <i>Class Diagram</i>	10
2.4.2 <i>Sequence Diagram</i>	10
2.4.3 <i>Use Case Diagram</i>	11
2.5 Teknologi Pengembangan Sistem.....	12
2.5.1 Hypertext Markup Language (HTML).....	12
2.5.2 <i>Cascading Style Sheet (CSS)</i>	13
2.5.3 <i>Bootstrap</i>	13
2.5.4 <i>Codeigniter</i>	13
2.5.5 PHP.....	14

2.5.6 JavaScript.....	14
2.6 Basis Data.....	14
2.6.1 Sistem Manajemen Basis Data.....	15
2.6.2 MySQL	15
2.7 Pengujian Perangkat Lunak.....	15
2.7.1 Pengujian Unit.....	16
2.7.2 Pengujian Integrasi.....	17
2.7.3 Pengujian Validasi	17
2.7.4 Pengujian <i>Compatibility</i>	17
BAB 3 METODOLOGI PENELITIAN	18
3.1 Tipe Penelitian	18
3.2 Strategi dan Rancangan Penelitian.....	18
3.2.1 Studi Literatur	19
3.2.2 Analisis Kebutuhan.....	19
3.2.3 Perancangan Sistem.....	19
3.2.4 Implementasi	19
3.2.5 Pengujian Sistem.....	20
3.2.6 Penarikan Kesimpulan.....	20
BAB 4 REKAYASA KEBUTUHAN.....	21
4.1 Elisitasi Kebutuhan.....	21
4.2 Deskripsi Sistem	23
4.3 Analisis Data.....	25
4.4 Identifikasi Pengguna.....	27
4.5 Identifikasi Kebutuhan.....	27
4.5.1 Kebutuhan Fungsional.....	27
4.5.2 Kebutuhan Non-Fungsional.....	35
4.6 Pemodelan Use Case.....	35
4.6.1 <i>Use Case Diagram</i>	35
4.6.2 <i>Use Case Scenario</i>	37
BAB 5 Perancangan dan implementasi	60
5.1 Perancangan	60
5.1.1 <i>Sequence Diagram</i>	60

5.1.2 Perancangan Basis Data	62
5.1.3 <i>Class Diagram</i>	63
5.1.4 Perancangan Algoritme.....	65
5.1.5 Perancangan Antarmuka.....	67
5.2 Implementasi	72
5.2.1 Spesifikasi Pengembangan Sistem	72
5.2.2 Implementasi Basis Data	72
5.2.3 Implementasi Algoritme.....	77
5.2.4 Implementasi Antarmuka	80
BAB 6 Pengujian	84
6.1 Pengujian Unit.....	84
6.1.1 Pengujian unit Fungsi Menyimpan Pendaftaran Berobat.....	84
6.1.2 Pengujian unit Fungsi Membuat Rekam Medis Baru.....	87
6.1.3 Pengujian unit Fungsi Menambah Data Rawat Jalan Gigi.....	90
6.2 Pengujian Validasi	92
6.3 Pengujian <i>Compatibility</i>	127
BAB 7 KESIMPULAN DAN SARAN	129
7.1 Kesimpulan.....	129
7.2 Saran	129
DAFTAR PUSTAKA.....	131
LAMPIRAN A KUNJUNGAN PASIEN BULAN FEBRUARI 2018	133
LAMPIRAN B FORMULIR IDENTITAS PASIEN BARU	135
LAMPIRAN C FORMULIR PENGKAJIAN KLINIS MEDIS RAWAT JALAN	136
LAMPIRAN D Formulir REKAM MEDIS UMUM.....	138
LAMPIRAN E FORMULIR REKAM MEDIS GIGI DAN MULUT	139
LAMPIRAN F FORMULIR PEMBERIAN INFORMASI DAN EDUKASI RAWAT JALAN.....	140
LAMPIRAN G FORMULIR PERMINTAAN PEMERIKSAAN DIAGNOSTIK	141
LAMPIRAN H HASIL WAWANCARA	142

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	5
Tabel 2.2 Isi Rekam Medis.....	6
Tabel 2.3 Isi Rekam Medis Elektronik	8
Tabel 4.1 Identifikasi Pengguna	27
Tabel 4.2 Daftar Kebutuhan Fungsional Sistem	28
Tabel 4.3 Pemetaan <i>Use Case</i>	33
Tabel 4.4 Daftar Kebutuhan Non-fungsional	35
Tabel 4.5 <i>Use Case Scenario</i> Daftar Berobat.....	37
Tabel 4.6 <i>Use Case Scenario</i> Log In	38
Tabel 4.7 <i>Use Case Scenario</i> Melihat <i>List</i> Pendaftaran Berobat.....	38
Tabel 4.8 <i>Use Case Scenario</i> Melihat Jumlah Kunjungan Rawat Jalan	39
Tabel 4.9 <i>Use Case Scenario</i> Melihat Peringkat Diagnosa	39
Tabel 4.10 <i>Use Case Scenario</i> Cetak Jumlah Kunjungan Rawat Jalan	40
Tabel 4.11 <i>Use Case Scenario</i> Cetak Peringkat Diagnosa	40
Tabel 4.12 <i>Use Case Scenario</i> Mengubah Status Konfirmasi Pendaftaran	41
Tabel 4.13 <i>Case Scenario</i> Membuat Rekam Medis Baru	41
Tabel 4.14 <i>Use Case Scenario</i> Melihat Data Pribadi Pasien	42
Tabel 4.15 <i>Use Case Scenario</i> Mengubah Data Pribadi Pasien	43
Tabel 4.16 <i>Use Case Scenario</i> Menghapus Rekam Medis.....	43
Tabel 4.17 <i>Use Case Scenario</i> Melihat <i>List</i> Pendaftaran Pasien Per Poli	44
Tabel 4.18 <i>Use Case Scenario</i> Mengubah Status Antrian Pasien.....	44
Tabel 4.19 <i>Use Case Scenario</i> Melihat Rekam Medis Pasien	45
Tabel 4.20 <i>Use Case Scenario</i> Melihat Riwayat Pengkajian Klinis Medis	46
Tabel 4.21 Melihat Detail Data Pengkajian Klinis Medis Rawat Jalan	46
Tabel 4.22 <i>Use Case Scenario</i> Membuat Pengkajian Klinis Medis Rawat Jalan Baru	47
Tabel 4.23 <i>Use Case Scenario</i> Membuat Data Rawat Jalan Umum Baru.....	47
Tabel 4.24 <i>Use Case Scenario</i> Membuat Data Rawat Jalan Gigi Baru	48
Tabel 4.25 <i>Use Case Scenario</i> Melihat Detail Data Rawat Jalan Umum	49
Tabel 4.26 <i>Use Case Scenario</i> Melihat Detail Data Rawat Jalan Gigi	49

Tabel 4.27 <i>Use Case Scenario</i> Menambah Data Rawat Jalan Umum	50
Tabel 4.28 <i>Use Case Scenario</i> Menambah Data Rawat Jalan Gigi	50
Tabel 4.29 <i>Use Case Scenario</i> Melihat Riwayat Kunjungan Umum	51
Tabel 4.30 <i>Use Case Scenario</i> Melihat Riwayat Kunjungan Gigi	51
Tabel 4.31 <i>Use Case Scenario</i> Melihat Riwayat Farmasi.....	52
Tabel 4.32 <i>Use Case Scenario</i> Menambah Data Terapi Farmasi.....	52
Tabel 4.33 <i>Use Case Scenario</i> Melihat Detail Terapi Farmasi	53
Tabel 4.34 <i>Use Case Scenario</i> Melihat Riwayat Diagnostik	54
Tabel 4.35 <i>Use Case Scenario</i> Menambah Permintaan Pemeriksaan Diagnostik	54
Tabel 4.36 <i>Use Case Scenario</i> Melihat Hasil Pemeriksaan Diagnostik Pasien	55
Tabel 4.37 <i>Use Case Scenario</i> Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan	55
Tabel 4.38 <i>Use Case Scenario</i> Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan	56
Tabel 4.39 <i>Use Case Scenario</i> Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan	57
Tabel 4.40 <i>Use Case Scenario</i> Melihat Daftar Permintaan Pemeriksaan Diagnostik	57
Tabel 4.41 <i>Use Case Scenario</i> Melihat Detail Permintaan Pemeriksaan Diagnostik	58
Tabel 4.42 <i>Use Case Scenario</i> Menambah Hasil Pemeriksaan Diagnostik.....	58
Tabel 4.43 <i>Use Case Scenario</i> Log out.....	59
Tabel 5.1 <i>Pseudocode</i> Menyimpan Pendaftaran Berobat	65
Tabel 5.2 <i>Pseudocode</i> Membuat Rekam Medis Pasien Baru	66
Tabel 5.3 <i>Pseudocode</i> Menambah Data Rawat Jalan Gigi	66
Tabel 5.4 Komponen Perancangan Halaman Awal	68
Tabel 5.5 Komponen Perancangan Halaman <i>home</i> Petugas Pendaftaran	68
Tabel 5.6 Komponen Perancangan Halaman <i>Home</i> Perawat.....	69
Tabel 5.7 Komponen Perancangan Halaman Pencarian Rekam Medis	70
Tabel 5.8 Komponen Perancangan Halaman <i>Home</i> Petugas Laboratorium	71
Tabel 5.9 Implementasi Basis Data Tabel admin	73
Tabel 5.10 Implementasi Basis Data Tabel Data Pribadi Pasien	73
Tabel 5.11 Implementasi Basis Data Tabel Diagnostik	73

Tabel 5.12 Implementasi Basis Data Tabel Edukasi Rawat Jalan	74
Tabel 5.13 Implementasi Basis Data Tabel Farmasi	75
Tabel 5.14 Implementasi Basis Data Tabel Pendaftaran	75
Tabel 5.15 Implementasi Basis Data Tabel Pengkajian Klinis Medis	75
Tabel 5.16 Implementasi Basis Data Tabel Rawat Jalan	77
Tabel 5.17 Implementasi Algoritme Menyimpan Pendaftaran Berobat	78
Tabel 5.18 Implementasi Algoritme Membuat Rekam Medis Baru	79
Tabel 5.19 Implementasi Algoritme Menambah data rawat jalan gigi	80
Tabel 6.1 <i>Pseudocode</i> Menyimpan Pendaftaran Berobat	84
Tabel 6.2 Hasil Pengujian Unit Fungsi Menyimpan Pendaftaran Berobat	86
Tabel 6.3 <i>Pseudocode</i> Membuat Rekam Medis Baru	88
Tabel 6.4 Kasus Uji Membuat Rekam Medis Baru	89
Tabel 6.5 <i>Pseudocode</i> Menambah Data Rawat Jalan Gigi	90
Tabel 6.6 Kasus Uji Menambah Data Rawat Jalan Gigi	92
Tabel 6.7 Kasus Uji Menyimpan Pendaftaran Berobat	92
Tabel 6.8 Kasus Uji Menyimpan Pendaftaran Berobat Alternatif 1 : Validasi <i>form</i> gagal	93
Tabel 6.9 Kasus Uji Menyimpan Pendaftaran Berobat Alternatif 2 : Koneksi dengan basis data terputus	93
Tabel 6.10 Kasus Uji <i>Log in</i>	94
Tabel 6.11 Kasus Uji <i>Log in</i> Alternatif 1 : Validasi <i>form</i> gagal	94
Tabel 6.12 Kasus Uji <i>Log in</i> Alternatif 2 : <i>Username</i> atau <i>password</i> salah	95
Tabel 6.13 Kasus Uji Melihat <i>List</i> Pendaftaran Berobat	95
Tabel 6.14 Kasus Uji Melihat <i>List</i> Pendaftaran Berobat Alternatif 1 : Data pendaftaran berobat kosong	95
Tabel 6.15 Kasus Uji Melihat Jumlah Kunjungan Rawat Jalan	96
Tabel 6.16 Kasus Uji Melihat Jumlah Kunjungan Rawat Jalan Alternatif 1 : Data jumlah kunjungan rawat jalan kosong	96
Tabel 6.17 Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan	96
Tabel 6.18 Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan Alternatif 1 : Aktor menekan tombol “Batal” pada halaman layout untuk mencetak	97
Tabel 6.19 Kasus Uji Melihat Peringkat Diagnosa	97
Tabel 6.20 Kasus Uji Cetak Peringkat Diagnosa	98

Tabel 6.21 Kasus Uji Cetak Peringkat Diagnosa Alternatif 1 : Aktor menekan tombol “Batal” pada halaman layout untuk mencetak	98
Tabel 6.22 Kasus Uji Mengubah Status Konfirmasi Pendaftaran.....	99
Tabel 6.23 Kasus Uji Membuat Rekam Medis Baru	99
Tabel 6.24 Kasus Uji Membuat Rekam Medis Baru Alternatif 1 : Validasi <i>form</i> gagal	99
Tabel 6.25 Kasus Uji Membuat Rekam Medis Baru Alternatif 2 : Koneksi dengan basis data terputus.....	100
Tabel 6.26 Kasus Uji Melihat Data Pribadi Pasien.....	100
Tabel 6.27 Kasus Uji Melihat Data Pribadi Pasien Alternatif 1 : Data pencarian rekam medis tidak ditemukan	101
Tabel 6.28 Kasus Uji Mengubah Data Pribadi Pasien.....	101
Tabel 6.29 Kasus Uji Mengubah Data Pribadi Pasien Alternatif 1 : Validasi <i>form</i> gagal	102
Tabel 6.30 Kasus Uji Mengubah Data Pribadi Pasien Alternatif 2 : Koneksi dengan basis data terputus.....	102
Tabel 6.31 Kasus Uji Menghapus Rekam Medis	103
Tabel 6.32 Kasus Uji Menghapus Rekam Medis Alternatif 1 : Aktor menekan tombol “Batal” pada kotak dialog menghapus data	103
Tabel 6.33 Kasus Uji Menghapus Rekam Medis Alternatif 2 : Koneksi dengan basis data terputus.....	103
Tabel 6.34 Kasus Uji Melihat <i>List</i> Pendaftaran Pasien Per Poli.....	104
Tabel 6.35 Kasus Uji Mengubah Status Antrian Pasien	104
Tabel 6.36 Kasus Uji Melihat Rekam Medis Pasien.....	105
Tabel 6.37 Kasus Uji Melihat Rekam Medis Pasien Alternatif 1 : Data pencarian rekam medis tidak ditemukan	105
Tabel 6.38 Kasus Uji Melihat Riwayat Pengkajian Klinis Medis	106
Tabel 6.39 Kasus Uji Melihat Riwayat Pengkajian Klinis Medis Alternatif 1 : Data pengkajian klinis medis kosong.....	106
Tabel 6.40 Kasus Uji Melihat Detail Data Pengkajian Klinis Medis	106
Tabel 6.41 Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru	107
Tabel 6.42 Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru Alternatif 1 : Koneksi dengan basis data terputus	107
Tabel 6.43 Kasus Uji Membuat Data Rawat Jalan Umum Baru	108

Tabel 6.44 Kasus Uji Membuat Data Rawat Jalan Umum Baru Alternatif 1: Ada <i>field</i> pada <i>form</i> yang belum terisi	108
Tabel 6.45 Kasus Uji Membuat Data Rawat Jalan Umum Baru Alternatif 2: Koneksi dengan basis data terputus.....	109
Tabel 6.46 Kasus Uji Membuat Data Rawat Jalan Gigi Baru	109
Tabel 6.47 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 1: Ada <i>field</i> pada <i>form</i> yang belum terisi	109
Tabel 6.48 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 2: Format <i>file</i> foto salah atau ukuran terlalu besar	110
Tabel 6.49 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 3: Koneksi dengan basis data terputus.....	110
Tabel 6.50 Kasus Uji Melihat Detail Data Rawat Jalan Umum.....	111
Tabel 6.51 Kasus Uji Melihat Detail Data Rawat Jalan Gigi.....	111
Tabel 6.52 Kasus Uji Menambah Data Rawat Jalan Umum	112
Tabel 6.53 Kasus Uji Menambah Data Rawat Jalan Umum Alternatif 1 : Ada <i>field</i> pada <i>form</i> yang belum terisi	112
Tabel 6.54 Kasus Uji Menambah Data Rawat Jalan Umum Alternatif 2 : Koneksi dengan basis data terputus.....	112
Tabel 6.55 Kasus Uji Menambah Data Rawat Jalan Gigi	113
Tabel 6.56 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 1 : Ada <i>field</i> pada <i>form</i> yang belum terisi	113
Tabel 6.57 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 2 Format <i>file</i> foto salah atau ukuran terlalu besar	114
Tabel 6.58 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 3: Koneksi dengan basis data terputus.....	114
Tabel 6.59 Kasus Uji Melihat Riwayat Kunjungan Umum.....	115
Tabel 6.60 Kasus Uji Melihat Riwayat Kunjungan Umum Alternatif 1 : Data riwayat kunjungan masih kosong.....	115
Tabel 6.61 Kasus Uji Melihat Riwayat Kunjungan Gigi.....	115
Tabel 6.62 Kasus Uji Melihat Riwayat Kunjungan Gigi Alternatif 1: Data riwayat kunjungan masih kosong.....	116
Tabel 6.63 Kasus Uji Melihat Riwayat Farmasi	116
Tabel 6.64 Kasus Uji Melihat Riwayat Farmasi Alternatif 1 : Data riwayat farmasi masih kosong.....	117
Tabel 6.65 Kasus Uji Menambah Data Terapi Farmasi	117

Tabel 6.66 Kasus Uji Menambah Data Terapi Farmasi Alternatif 1: Validasi <i>form</i> gagal	117
Tabel 6.67 Kasus Uji Menambah Data Terapi Farmasi Alternatif 2 : Koneksi dengan basis data terputus.....	118
Tabel 6.68 Kasus Uji Melihat Detail Terapi Farmasi.....	118
Tabel 6.69 Kasus Uji Melihat Riwayat Diagnostik	119
Tabel 6.70 Kasus Uji Melihat Riwayat Diagnostik Alternatif 1: Data riwayat diagnostik masih kosong	119
Tabel 6.71 Kasus Uji Melihat Hasil Pemeriksaan Diagnostik	120
Tabel 6.72 Kasus Uji Menambah Permintaan Pemeriksaan Diagnostik	120
Tabel 6.73 Kasus Uji Menambah Permintaan Pemeriksaan Diagnostik Alternatif 1 : Koneksi dengan basis data terputus	121
Tabel 6.74 Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan.....	121
Tabel 6.75 Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 1 : Data riwayat edukasi rawat jalan masih kosong	121
Tabel 6.76 Kasus Uji Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan.....	122
Tabel 6.77 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan.....	122
Tabel 6.78 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 1 : Validasi <i>form</i> gagal.....	123
Tabel 6.79 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 2 : Koneksi dengan basis data terputus.....	123
Tabel 6.80 Kasus Uji Melihat Daftar Permintaan Pemeriksaan Diagnostik	124
Tabel 6.81 Kasus Uji Melihat Daftar Permintaan Pemeriksaan Diagnostik Alternatif 1 : Data permintaan pemeriksaan diagnostik kosong	124
Tabel 6.82 Kasus Uji Melihat Detail Data Melihat Detail Permintaan Pemeriksaan Diagnostik.....	125
Tabel 6.83 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik	125
Tabel 6.84 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik Alternatif 1 : Validasi <i>form</i> gagal	125
Tabel 6.85 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik Alternatif 2 : Format <i>file</i> foto salah atau ukuran <i>file</i> terlalu besar	126
Tabel 6.86 Kasus Uji Menambah Menambah Hasil Pemeriksaan Diagnostik Alternatif 3 : Koneksi dengan basis data terputus.....	126

Tabel 6.87 Kasus Uji <i>Log out</i>	127
---	-----



DAFTAR GAMBAR

Gambar 2.1 Model Waterfall	9
Gambar 2.2 Tahapan Proses Analisis dan Perancangan Berorientasi Objek	10
Gambar 2.3 Ilustrasi Kotak dalam Class Diagram	10
Gambar 2.4 Ilustrasi <i>Sequence Diagram</i>	11
Gambar 2.5 Contoh <i>Use Case Diagram</i>	12
Gambar 2.6 Notasi <i>Flow Graph</i>	16
Gambar 3.1 Tahapan Metodologi Penelitian	18
Gambar 4.1 Alur Pelayanan Pasien Klinik Medis Elisa Saat Ini	22
Gambar 4.2 Alur Sistem yang Ditawarkan	24
Gambar 4.3 <i>Entity Relationship Diagram</i>	26
Gambar 4.4 <i>Use Case Diagram</i>	36
Gambar 5.1 <i>Sequence Diagram</i> Menyimpan Pendaftaran Berobat	60
Gambar 5.2 <i>Sequence Diagram</i> Membuat Rekam Medis Baru	61
Gambar 5.3 <i>Sequence Diagram</i> Menambah Data Rawat Jalan Gigi	62
Gambar 5.4 <i>Physical Data Model</i>	63
Gambar 5.5 Class Diagram	64
Gambar 5.6 Perancangan Antarmuka Halaman Awal	67
Gambar 5.7 Perancangan Antarmuka Halaman <i>Home</i> Petugas Pendaftaran	68
Gambar 5.8 Perancangan Antarmuka Halaman <i>Home</i> Perawat	69
Gambar 5.9 Perancangan Antarmuka Halaman Pencarian Rekam Medis	70
Gambar 5.10 Perancangan Halaman Antarmuka Halaman <i>Home</i> Petugas Laboratorium	71
Gambar 5.11 Implementasi Halaman Awal	81
Gambar 5.12 Implementasi Halaman <i>Home</i> Petugas Pendaftaran	81
Gambar 5.13 Implementasi Halaman Pencarian Rekam Medis	82
Gambar 5.14 Implementasi Halaman <i>Home</i> Perawat	82
Gambar 5.15 Implementasi Halaman <i>Home</i> Petugas Laboratorium	83
Gambar 6.1 <i>Flow Graph</i> Daftar Berobat Pasien Lama	85
Gambar 6.2 <i>Flow Graph</i> Membuat Rekam Medis Baru	88
Gambar 6.3 <i>Flow Graph</i> Menambah Data Rawat Jalan Gigi	91

Gambar 6.4 Hasil Pengujian <i>Compatibility</i>	128
---	-----



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Manajemen pengelolaan informasi adalah hal utama yang harus dipenuhi oleh suatu instansi atau perusahaan, khususnya bila memiliki data yang besar dan kompleks. Pada era digital ini, telah banyak perusahaan dan instansi yang mengandalkan sistem informasi sebagai alat untuk membantu mengelola manajemen informasi dan data penting yang dimiliki. Sistem informasi menurut Ian Sommerville (2011) adalah sistem yang memiliki tujuan utama untuk mengelola dan menyediakan akses ke basis data. Sistem informasi juga berperan penting dalam distribusi informasi dan sebagai alat bantu untuk mengambil keputusan. Di Indonesia, sistem informasi telah banyak digunakan pada berbagai bidang, tak terkecuali bidang kesehatan. Sistem informasi dalam bidang kesehatan adalah seperangkat tatanan yang meliputi data, informasi, indikator, prosedur, perangkat, teknologi, dan sumber daya manusia yang saling berkaitan secara terpadu untuk mengarahkan tindakan atau keputusan yang berguna dalam mendukung pembangunan kesehatan (Peraturan Pemerintah nomor 46 tahun 2014).

Saat ini telah banyak rumah sakit, klinik, puskesmas, maupun praktik dokter yang telah menggunakan sistem informasi untuk memudahkan manajemen data dan meningkatkan mutu pelayanan kepada pasien. Klinik Elisa yang berlokasi di jalan Danau Toba E5/ 22 Malang juga sangat peduli dengan kepuasan pasien. Klinik yang berdiri sejak tahun 2007 ini ingin selalu memberikan pelayanan terbaik setiap saat. Klinik Medis Elisa menyediakan beberapa layanan kesehatan, antara lain pelayanan umum, pelayanan gigi, unit gawat darurat, kesehatan ibu dan anak, serta rawat inap, yang dilengkapi laboratorium dan apotek. Klinik Medis Elisa juga melayani pasien BPJS (Badan Penyelenggara Jaminan Sosial) Kesehatan di tingkat pertama. Pasien di klinik ini digolongkan menjadi dua jenis yaitu mandiri dan asuransi.

Pada alur pemeriksaan pasien rawat jalan, prosedur pertama yang harus dilalui adalah pendaftaran. Pendaftaran pasien dilakukan di loket dan dilayani oleh 3 orang petugas pendaftaran dan 1 orang yang bertugas menyiapkan *file* rekam medis karena Klinik Medis Elisa masih menggunakan rekam medis berupa *hard file* yang ditulis oleh dokter pemeriksa yang kemudian disimpan dalam rak-rak khusus. Namun prosedur ini sering mengalami kendala pada jumlah pasien yang sangat banyak dan tidak diimbangi dengan jumlah tenaga kerja yang mencukupi. Pada bulan Februari 2018, jumlah kunjungan pasien BPJS mencapai angka 2.638, sedangkan kunjungan pasien mandiri sebanyak 325 kunjungan untuk pemeriksaan pelayanan umum, dan 93 kunjungan pasien laboratorium. Akibatnya, prosedur pendaftaran memakan waktu yang lama. Satu orang pasien bisa memakan waktu 5-15 menit untuk melakukan proses pendaftaran, termasuk pencarian rekam medis pasien atau pembuatan dokumen rekam medis untuk pasien yang pertama kali berobat. Proses ini dapat memakan waktu lebih lama lagi jika dokumen rekam

medis pasien ternyata tidak ditemukan atau ditemukan dalam keadaan rusak, petugas harus membuat rekam medis yang baru.

Karena itu, untuk mempersingkat waktu kunjungan pasien, diperlukan pendaftaran pasien secara *online* yang dapat diakses oleh pasien dimanapun dari *gadget* mereka sehingga tidak perlu datang lebih awal ke klinik untuk mengantri di loket pendaftaran. Hal ini akan lebih menghemat waktu pasien dan meringankan kerja petugas pendaftaran di loket. Selain itu, merujuk pada penelitian yang dilakukan Tiara Handayani dan Gerson Feoh dengan judul “Perancangan Sistem Informasi Rekam Medis Berbasis Web Studi Kasus di Klinik Bersalin Sriati kota Sungai Penuh, Jambi” yang dilakukan pada tahun 2016, dengan adanya sistem rekam medis elektronik, antrian pasien saat registrasi dapat dikurangi dan pelayanan pasien akan lebih maksimal dengan adanya sistem yang mengelola *file* rekam medis yang telah diubah kedalam bentuk elektronik sehingga lebih mudah dalam pencarian, meminimalisasi sumber daya manusia yang digunakan, dan menghindari kerusakan/kehilangan dokumen rekam medis pasien (Handayani & Feoh, 2016).

Dalam membangun sebuah perangkat lunak yang sesuai dengan *Software Development Life Cycle* (SDLC), terdapat empat tahap yaitu analisis kebutuhan, perancangan, implementasi, dan pengujian. Analisis kebutuhan dapat dilakukan dengan mempertemukan klien dan tim pengembang perangkat lunak untuk membahas tujuan pembuatan perangkat lunak, apa tipe data yang digunakan, bagaimana perangkat lunak akan mengelola data, dan bagaimana data akan diakses dalam sistem. Pada tahap perancangan, hasil dari analisis kebutuhan akan diterjemahkan ke rencana perancangan perangkat lunak, kemudian hasil dari perancangan akan diterjemahkan ke dalam kode program pada tahap implementasi. Program yang telah jadi akan diuji dengan serangkaian pengujian untuk verifikasi bahwa perangkat lunak dapat memenuhi tujuan yang telah didefinisikan pada analisis kebutuhan. Pada penelitian ini digunakan pemodelan kebutuhan dan perancangan berorientasi objek yaitu dekomposisi arsitektur sistem berdasarkan objek-objek yang dimanipulasi oleh sistem yang sedang dibangun.

Mengacu pada masalah sebelumnya, didapatkan judul penelitian “Pembangunan Sistem Aplikasi Rekam Medis Elektronik dan Pendaftaran Pasien *Online* Berbasis Web” dengan studi kasus Klinik Elisa Malang. Dengan adanya sistem ini diharapkan manajemen rekam medis pasien rawat jalan menjadi lebih teratur dan memberikan kemudahan kepada pasien dalam hal pendaftaran sehingga dapat mempersingkat waktu kunjungan berobat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka diperoleh rumusan masalah sebagai berikut :

1. Bagaimana hasil analisis kebutuhan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang ?
2. Bagaimana hasil perancangan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang ?

3. Bagaimana hasil implementasi sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang ?
4. Bagaimana hasil pengujian dari sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang ?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah diuraikan, maka diperoleh tujuan penelitian sebagai berikut :

1. Mengetahui hasil analisis kebutuhan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang.
2. Mengetahui hasil perancangan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang.
3. Mengetahui hasil implementasi sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web di Klinik Elisa Malang.
4. Mengetahui hasil pengujian dari sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web Klinik Elisa Malang.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi penulis dan instansi terkait, yaitu :

1. Bagi Penulis
Dengan dibangunnya sistem rekam medis elektronik dan pendaftaran pasien *online* berbasis web, diharapkan dapat menambah wawasan dan mengimplementasikan ilmu yang didapat agar berguna bagi kepentingan masyarakat.
2. Bagi Klinik Medis Elisa Malang
Dengan dibangunnya sistem rekam medis elektronik dan pendaftaran pasien *online* berbasis web diharapkan dapat meningkatkan kualitas pelayanan kepada pasien dengan alat bantu sistem informasi yang telah dibuat dan meningkatkan minat pasien untuk berobat karena dimudahkan dengan sistem informasi yang telah dibuat.

1.5 Batasan Masalah

Agar penelitian dapat dilakukan dengan lebih fokus, maka dibuat batasan masalah sebagai berikut:

1. Sistem rekam medis dibuat untuk pemeriksaan rawat jalan pada Klinik Medis Elisa.
2. Sistem rekam medis digunakan oleh petugas yang berhubungan dengan rekam medis pasien rawat jalan yaitu petugas pendaftaran, petugas rekam medis, perawat, dan petugas laboratorium. Pendaftaran *online* ditujukan untuk pasien.
3. Data penelitian untuk membangun sistem diperoleh dari observasi langsung di lapangan dan wawancara dengan pihak Klinik Medis Elisa.

4. Analisis, perancangan, dan implementasi sistem dilakukan dengan pendekatan berorientasi objek.
5. *Software Development Life Cycle* yang digunakan dalam pembangunan sistem menggunakan model *Waterfall*.
6. Bahasa pemrograman yang digunakan untuk membangun sistem adalah PHP dan JavaScript.
7. Sistem dibangun adalah berbasis web dan *framework* yang digunakan untuk membangun sistem adalah Codeigniter dan Bootstrap.
8. *Code editor* yang digunakan untuk membangun sistem adalah Sublime Text 3.
9. Basis data yang digunakan adalah MySQL.

1.6 Sistematika Pembahasan

Sistematika penulisan ditujukan untuk memberi gambaran dari seluruh skripsi ini yaitu sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat rumusan masalah yang terkait dengan hal – hal yang ingin dikaji dalam penelitian , tujuan dilakukan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan skripsi.

BAB II LANDASAN KEPUSTAKAAN

Bab ini menjelaskan kajian pustaka serta teori penunjang yang mendasari penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metode atau langkah – langkah yang dilakukan dalam penelitian skripsi, yaitu studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta penarikan kesimpulan dan saran.

BAB IV REKAYASA KEBUTUHAN

Bab ini berisi tahap-tahap untuk menemukan dan menganalisis kebutuhan-kebutuhan untuk perancangan dan pembangunan perangkat lunak.

BAB V PERANCANGAN DAN IMPLEMENTASI

Bagian ini memuat perancangan sistem berdasarkan rekayasa kebutuhan pada bab sebelumnya dan implementasinya.

BAB VI PENGUJIAN

Bab ini berisi strategi dan teknik pengujian aplikasi yang telah dibangun, kasus uji beserta hasil pengujian yang diperoleh.

BAB VII PENUTUP

Bab ini memuat kesimpulan yang merupakan jawaban dari rumusan masalah yang diperoleh hasil penelitian, dan saran untuk pengembangan penelitian lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Penelitian ini dilakukan dengan merujuk pada beberapa penelitian terdahulu. Penelitian pertama mempunyai judul “Perancangan Sistem Informasi Rekam Medis Rawat Inap Rumah Sakit Berbasis Web” yang disusun oleh Nunik Srikandi Putri, Adian Fatchur Rochim, dan Ike Pertiwi Windasari dalam Jurnal Teknologi dan Sistem Komputer Universitas Diponegoro. Penelitian tersebut dilakukan untuk mengelola rekam medis khususnya rekam medis rawat inap pada sebuah rumah sakit agar menjadi lebih efisien dengan membangun sistem informasi berbasis web. Hasil dari penelitian ini yaitu dengan adanya sistem informasi rekam medis pasien rawat inap, pengelolaan data menjadi lebih efisien dan juga data tersimpan dengan aman.

Penelitian kedua yaitu berjudul Perancangan Sistem Informasi Rekam Medis Berbasis Web (Studi Kasus Di Klinik Bersalin Sriati Kota Sungai Penuh – Jambi) yang ditulis oleh Tiara Handayani dan Gerson Feoh dalam Jurnal Undhira Bali. Penelitian ini bertujuan untuk memudahkan petugas rekam medis dalam mengelola data pasien yaitu pendaftaran pasien, pencatatan rekam medis pasien, pencatatan data dokter, dan data ruang. Hasil penelitian ini menunjukkan bahwa dengan adanya sistem informasi rekam medis berbasis web dapat mengurangi antrean pasien saat registrasi dan memudahkan petugas dalam pencatatan rekam medis.

Penelitian-penelitian rujukan yang telah dijelaskan sebelumnya dapat dilihat secara rinci pada Tabel 2.1

Tabel 2.1 Tinjauan Pustaka

No	Judul dan Tahun	Penulis	Hasil Penelitian	Penelitian Penulis
1	Perancangan Sistem Informasi Rekam Medis Rawat Inap Rumah Sakit Berbasis Web [2014]	Nunik Srikandi Putri, Adian Fatchur Rochim, Ike Pertiwi Windasari	Dengan adanya Sistem Informasi Rekam Pasien Medis Rawat Inap, pengelolaan data rekam medis menjadi lebih mudah, dan efisien. Data rekam medis aman karena tersimpan dalam database. Lebih terstruktur dibandingkan pengelolaan secara manual.	Implementasi Sistem Informasi Rekam Medis Pasien Rawat Inap dibangun menggunakan web server Apache HTTP Server 2.2.4, bahasa pemrograman PHP, dan framework Codeigniter. Sistem dibuat tersentralisasi dengan meletakkan aplikasi dan basis data pada server tunggal.
2	Perancangan Sistem Informasi Rekam Medis Berbasis Web (Studi Kasus Di Klinik Bersalin Sriati Kota	Tiara Handayani, Gerson Feoh	Sistem rekam medis elektronik dapat mengurangi antrian pasien saat registrasi, memudahkan petugas	Metode penelitian menggunakan System Development Life Cycle (SDLC) dengan jenis

	Sungai Penuh – Jambi) [2016]		dalam mencatat rekam medis pasien, dan menghasilkan laporan administrasi yang digunakan untuk kebutuhan manajemen klinik bersalin.	pendekatan waterfall. Perancangan sistem menggunakan pendekatan terstruktur. Implementasi sistem menggunakan bahasa pemrograman PHP.
--	------------------------------	--	--	--

2.2 Rekam Medis

Menurut Manual Rekam Medis yang dikeluarkan oleh Konsil Kedokteran Indonesia, salah satu unsur utama yang harus dipenuhi oleh dokter dan dokter gigi sebagai salah satu komponen utama pemberi pelayanan kesehatan kepada masyarakat adalah tersedianya rekam medis. Rekam medis harus terjaga kualitasnya dan sesuai dengan amanah Undang-Undang Nomor 29 Tahun 2004 tentang Praktik Kedokteran. Dalam penyelenggaraan praktik kedokteran, setiap dokter dan dokter gigi wajib mengacu pada standar, pedoman dan prosedur yang berlaku sehingga masyarakat mendapat pelayanan medis secara profesional dan aman. Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 269 Bab I, Pasal 1 Ayat (1), rekam medis adalah berkas yang berisi dokumen dan catatan tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien (Pramesti, 2018).

Saat ini sarana pelayanan kesehatan di Indonesia wajib mengadakan penyelenggaraan rekam medis dengan tujuan supaya administrasi sarana pelayanan kesehatan seperti klinik, puskesmas, atau rumah sakit menjadi lebih tertib. Manfaat rekam medis yang tercantum pada Pasal 13 Bab V Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/Menkes/Per/11/2008 Tentang Rekam Medis yaitu : sebagai alat untuk pemeliharaan dan pengobatan pasien, alat bukti dalam proses penegakan hukum, disiplin, dan etika kedokteran dan kedokteran gigi. Rekam medis juga digunakan sebagai dasar dalam pembayaran biaya kesehatan dan sebagai data statistik kesehatan.

Pengisian rekam medis dilakukan oleh dokter, dokter gigi, dan / atau petugas kesehatan tertentu, sedangkan penjelasan isi rekam medis hanya boleh dilakukan dokter atau dokter gigi yang menangani pasien. Isi rekam medis untuk pasien rawat jalan seperti yang tertulis pada Bab II Pasal 3 ayat (1) Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/Menkes/Per/11/2008 Tentang Rekam Medis yaitu terdiri dari beberapa komponen seperti yang terdapat pada Tabel 2.2.

Tabel 2.2 Isi Rekam Medis

No	Nama komponen	Ketentuan
1	Identitas pasien	<ul style="list-style-type: none"> Informasi personal pasien seperti nama, alamat, nomor telepon yang dapat dihubungi, etnis, bahasa dan lain-lain. Nama pasien harus ada pada tiap halaman

2	Tanggal dan waktu	Tanggal dan waktu dari pelayanan harus ada pada semua entri
3	Hasil anamnesis	Hasil anamnesis berisi keluhan, gejala dan riwayat penyakit pasien
4	Hasil pemeriksaan fisik dan penunjang medik	<ul style="list-style-type: none"> • Pemeriksaan fisik meliputi pemeriksaan tekanan darah, pernafasan, suhu badan, denyut jantung, tinggi badan, berat badan, pemeriksaan indera dan kelenjar • Penunjang medik yaitu riwayat pemeriksaan laboratorium, radiologi, dan penunjang medik lainnya
5	Diagnosis	<ul style="list-style-type: none"> • Diagnosis ditentukan berdasarkan hasil pemeriksaan fisik dan penunjang medik • Perkembangan pasien, tanggapan dan perubahan terapi, dan revisi dari diagnosis harus didokumentasikan • Diagnosa sebelum dan diagnosa saat ini harus dapat diakses oleh petugas pelayanan medis
6	Rencana penatalaksanaan	Rencana penatalaksanaan berisi rencana penanganan penyakit pasien yang akan dilakukan dokter
7	Pengobatan dan / atau tindakan	<ul style="list-style-type: none"> • Pengobatan pasien terdiri dari nama obat, dosis, frekuensi, tanggal dari awal dan/atau pengulangan resep, pengobatan herbal, suplemen, dan lain-lain • Tindakan terdiri dari semua tindakan medis yang diberikan kepada pasien
8	Pelayanan lain yang diberikan kepada pasien	Pelayanan berupa pencatatan edukasi, penyuluhan, pencatatan alergi pasien, persetujuan atau penolakan pasien
9	Odontogram	<ul style="list-style-type: none"> • Odontogram wajib ada untuk pasien kasus gigi • Odontogram adalah visualisasi susunan gigi manusia
10	Persetujuan tindakan	Persetujuan tindakan yang dilakukan petugas medis kepada pasien

Terdapat dua jenis rekam medis yaitu rekam medis yang ditulis secara lengkap (*paper-based*) dan rekam medis elektronik yang dibuat dengan teknologi informasi.

2.2.1 Rekam Medis Elektronik

Rekam Medis Elektronik atau *Electronic Medical Record (EMR)* adalah istilah umum untuk sistem informasi kesehatan terpadu yang berbasis komputer dan dapat diakses ditempat pasien melakukan perawatan (Scott, et al., 2007). EMR adalah sebuah dokumen yang multifungsi, didalamnya terdapat catatan elektronik kesehatan pasien yang diperbaharui secara *real time*. Informasi dimasukkan pada komputer kerja atau antarmuka yang saling terhubung. Dibanding dengan rekam medis manual, EMR lebih unggul karena sangat meningkatkan akurasi perekaman informasi pasien untuk memfasilitasi proses diagnosa. EMR juga menyediakan *platform* untuk memelihara dan membaca

informasi kesehatan pasien secara terus menerus. Dengan adanya EMR, dapat memperpanjang umur data pasien dengan menerapkan alat manajemen informasi.

Komponen isi rekam medis elektronik menurut Scott, et al (2007) seperti tertulis pada Tabel 2.3.

Tabel 2.3 Isi Rekam Medis Elektronik

No	Fungsi/Aplikasi	Deskripsi
1	Urutan entri oleh petugas kesehatan : <ul style="list-style-type: none"> • Manajemen laboratorium • Manajemen farmasi • Manajemen pencitraan diagnostik • Manajemen rujukan • Pendukung keputusan • Peringatan 	Untuk mendukung pemesanan pemeriksaan laboratorium, pencitraan diagnostik, atau permintaan konsultasi.
2	Catatan Pasien Elektronik	Penyimpanan yang terintegrasi dan presentasi dari informasi pasien
3	Manajemen dokumen	Mengijinkan petugas kesehatan untuk merekam tindakan yang telah dilakukan dalam mendiagnosis dan merawat pasien. Hal ini mencakup catatan kemajuan dari dokter dan perawat dan catatan administrasi obat.
4	Pendukung Keputusan Klinis	Lansiran yang didasarkan pada data saat ini dari catatan medis elektronik, pedoman praktik, atau sistem kecerdasan buatan yang lebih kompleks untuk dukungan diagnostik yang diberikan pada saat petugas kesehatan sedang membuat penilaian kondisi pasien dan membuat keputusan.
5	Data Administratif	Akses ke data administratif seperti catatan masuk, keluar dan transfer, jadwal operasi, data demografis, data ketersediaan ruang, dan lain-lain.
6	Pendukung Komunikasi Terintegrasi	Alat-alat yang memfasilitasi komunikasi yang efektif dan efisien antara anggota tim (termasuk pasien) untuk mendukung kelangsungan perawatan.
7	Akses ke Sumber Pengetahuan	Informasi <i>online</i> , termasuk bahan referensi, artikel jurnal, pedoman, dan lain-lain, pada saat keputusan dibuat mengenai perawatan pasien.

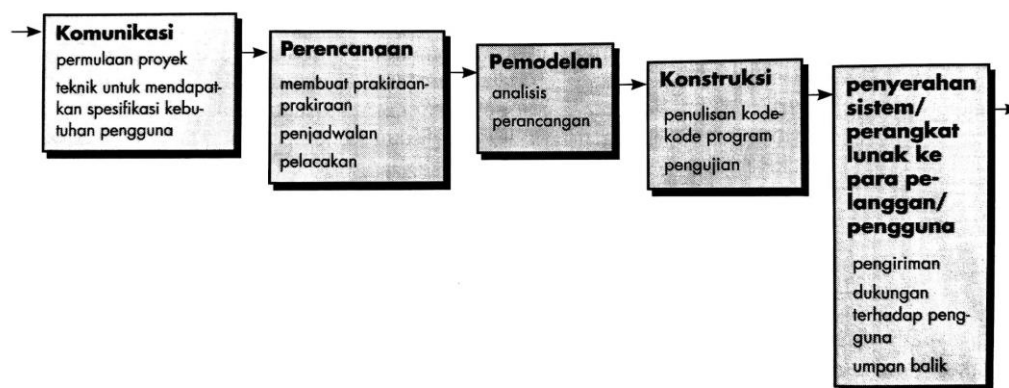
2.3 System Development Life Cycle

Siklus hidup pengembangan sistem atau disebut *System Development Life Cycle* (SDLC) secara keseluruhan mencakup empat komponen. Menurut Langer (2016), pengembangan sistem terdiri dari semua langkah yang diperlukan untuk mencapai pembangunan aplikasi, yaitu : kelayakan (*feasibility*), analisis, desain, dan pengkodean. Kelayakan diperlukan untuk menentukan apakah proyek perangkat lunak masuk pada kegiatan bisnis. Analisis adalah langkah untuk membuat kebutuhan yang logis dan terperinci untuk pembangunan sistem. Desain adalah fase yang membutuhkan keputusan tentang fisik sistem, dari bahasa pemrograman apa yang digunakan, vendor basis data mana yang harus dipilih (misalnya, Oracle, Sybase, DB2), hingga bagaimana laporan akan diidentifikasi. Pengkodean merepresentasikan arsitektur serta pendekatan matematis. Pengkodean adalah cara terbaik untuk benar-benar mengembangkan struktur program.

Terdapat beberapa pendekatan dalam SDLC yang dapat digunakan untuk membangun sebuah sistem, yaitu *Code-and-fix model*, *Waterfall model*, *Spiral model*, *Staged Delivery model*, dan *Evolutionary Prototyping model*. Masing-masing pendekatan memiliki ciri khas, keunggulan, dan batasan dalam proses pembangunan perangkat lunak.

2.3.1 Model Waterfall

Model *Waterfall* atau dapat juga disebut *classic life cycle* adalah siklus hidup perangkat lunak yang menggunakan pendekatan sistematis dan sekuensial untuk pengembangan perangkat lunak yang terdiri dari beberapa tahap yang diawali dengan spesifikasi kebutuhan dari pelanggan kemudian dikembangkan melalui perencanaan, pemodelan, konstruksi, dan distribusi, yang berpuncak pada pemeliharaan berkelanjutan dari perangkat lunak yang telah selesai (Pressman & Maxim, 2015). Model ini cocok digunakan bila pengembang telah atau pernah mengembangkan perangkat lunak yang serupa sebelumnya dan sadar akan setiap permasalahan yang ada dalam proses pengembangan perangkat lunak.



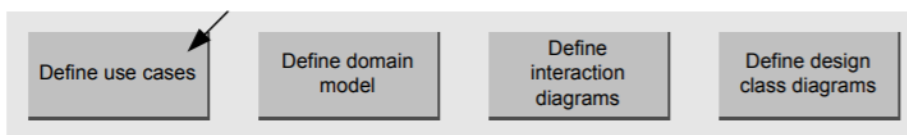
Gambar 2.1 Model Waterfall

Sumber : Pressman & Maxim (2015)

2.4 UML (Unified Modeling Language)

Unified Modeling Language atau biasa disingkat dengan UML adalah bahasa untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak (Larman, 2001). UML juga digunakan untuk memodelkan bisnis dan sistem non-perangkat lunak. UML adalah bahasa pemodelan yang berorientasi objek. Sistem dimodelkan sebagai kumpulan objek diskrit yang saling berinteraksi untuk menjalankan perintah dari user.

Membangun sistem yang menggunakan analisis dan perancangan yang berorientasi objek terdiri dari beberapa proses yaitu mendefinisikan *use case*, mendefinisikan *domain model*, mendefinisikan diagram yang berisi interaksi dalam sistem, dan mendefinisikan perancangan *class diagram*.

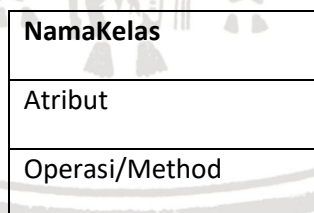


Gambar 2.2 Tahapan Proses Analisis dan Perancangan Berorientasi Objek

Sumber : Larman (2001)

2.4.1 Class Diagram

Diagram kelas atau *class diagram* digunakan untuk menampilkan kelas-kelas yang ada dalam sistem dan relasi antarkelas dalam sistem. Diagram ini merupakan representasi dari sistem secara statis (Larman, 2001). Sebuah kotak *class diagram* yang mengilustrasikan sebuah kelas dalam sistem memiliki tiga bagian yaitu nama *class*, atribut, dan operasi/method. Atribut dan operasi kelas dapat memiliki tiga sifat yaitu : *public* yang dapat dipanggil oleh kelas apa saja, *protected* yang dapat dipanggil oleh class tersebut dan turunannya, dan *private* yang hanya dapat dipanggil di kelas itu sendiri.



Gambar 2.3 Ilustrasi Kotak dalam Class Diagram

Sumber : Larman (2001)

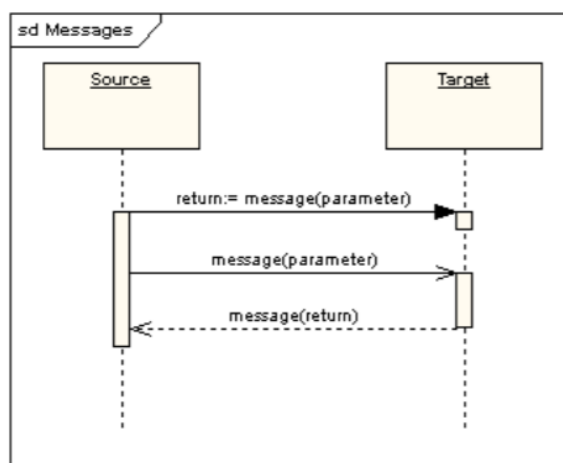
2.4.2 Sequence Diagram

Sequence Diagram dari sebuah sistem adalah diagram yang menunjukkan skenario tertentu dari sebuah *use case*, peristiwa yang dihasilkan oleh aktor yang berinteraksi dengan sistem, perintah/pesanan dari aktor, dan peristiwa yang terjadi antarsistem (Larman, 2001). *Sequence diagram* digunakan untuk

menunjukkan objek mana yang berkomunikasi dengan objek lainnya, dan pesan apa yang yang memicu komunikasi tersebut.

Komponen dari *sequence diagram* menurut Spark System (2007) adalah *lifeline* dan pesan. Sebuah *lifeline* adalah partisipan individu pada *sequence diagram*. Sebuah *lifeline* biasanya digambarkan dengan persegi panjang yang didalamnya berisi nama objek. Kadang-kadang *sequence diagram* memiliki *lifeline* dengan simbol elemen aktor. Hal ini terjadi jika *sequence diagram* dimiliki oleh sebuah *use case*. *Boundary*, *control*, dan *entity* juga dapat memiliki *lifeline*.

Komponen pesan atau *message* digambarkan dalam bentuk panah. Pesan dapat berupa pesan utuh, *lost or found*, *synchronous* atau *asynchronous*, *call* atau *signal*. Pada gambar 2.4, pesan pertama adalah pesan *synchronous* (notasi garis dengan anak panah padat) yang dilengkapi dengan *return message*. Pesan kedua adalah *asynchronous* (notasi garis dengan anak panah), kemudian pesan ketiga adalah *asynchronous return message* (notasi garis putus-putus).



Gambar 2.4 Ilustrasi *Sequence Diagram*

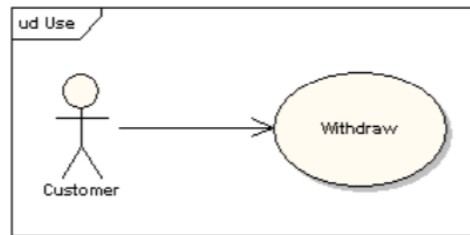
Sumber : Spark Systems (2007)

2.4.3 Use Case Diagram

Use case diagram menyediakan diagram konteks visual yang ringkas untuk sistem, menggambarkan aktor eksternal dan bagaimana mereka menggunakan sistem (Larman, 2001). *Use case diagram* juga menggambarkan apa saja yang dilakukan sistem atau subsistem namun tidak menjelaskan spesifikasi cara kerjanya.

Sebuah *use case diagram* menunjukkan interaksi antara sistem dengan entitas eksternal yang berhubungan dengan sistem. *Use case diagram* memiliki beberapa komponen yaitu : aktor, *use case*, batas, dan hubungan antara aktor dan *use case*. Aktor adalah manusia, perangkat keras lain, atau sistem lain yang berhubungan dengan sistem yang dikembangkan. Notasi aktor digambarkan dengan *stick man* yang memiliki nama. *Use case* adalah sebuah unit yang menggambarkan pekerjaan yang dilakukan sistem. Notasi untuk *use case* adalah *ellipse*. Notasi untuk menggunakan *use case* adalah sebuah garis penghubung

dengan anak panah (opsional) yang menunjukkan arah kontrol. Diagram pada gambar 2.4 menunjukkan bahwa aktor “customer” menggunakan *use case* “withdraw”.



Gambar 2.5 Contoh Use Case Diagram

Sumber : Spark Systems (2007)

2.5 Teknologi Pengembangan Sistem

2.5.1 Hypertext Markup Language (HTML)

Dokumen HTML yang juga dikenal sebagai halaman web adalah dokumen yang berisi informasi atau *interface* dari sebuah aplikasi di dalam internet (Sidhik & Pohan, 2014). Terdapat dua cara untuk membuat sebuah halaman web, yaitu dengan HTML *editor* atau dengan *editor* teks biasa. Untuk penamaan dokumen HTML, dapat memilih sembarang namap kemudian menambahkan ekstensi “.htm” atau “.html”.

Membuat dokumen HTML relatif mudah. Salah satu sifat HTML yaitu HTML adalah berupa teks biasa. Konten dan tanda (*tag*) yang ada didalamnya juga termasuk teks sehingga HTML dapat ditulis di editor teks apa pun. Jika menggunakan Windows, maka dapat menggunakan editor teks Notepad (Pfaffenberger, et al., 2004).

Dokumen HTML disusun dari elemen-elemen. Elemen yang dibutuhkan untuk membuat suatu dokumen HTML dinyatakan dengan *tag* <html>, <head>, dan <body> berikut *tag-tag* pasangannya. Setiap dokumen terdiri dari *tag head* dan *body*. Elemen *head* berisi informasi tentang dokumen tersebut, dan elemen *body* berisi teks yang tersusun dari *link*, grafik, paragraf dan elemen lainnya.

Setiap dokumen HTML harus mempunyai pola dasar sebagai berikut :

```

<html>
<head>
.....informasi tentang dokumen HTML
</head>
<body>
.....informasi yang ditampilkan dalam web browser
</body>
</html>
  
```

Dokumen HTML diawali dengan menulis *tag* <html> dan diakhiri dengan *tag* </html> di akhir dokumen. Di dalam satu dokumen hanya ada satu elemen HTML. Elemen *head* ditandai dengan *tag* <head> dan diakhiri dengan *tag* </head>. Minimal informasi yang terdapat pada bagian ini berisi informasi judul dari dokumen yang akan ditampilkan pada *caption bar* pada *web browser*. Bagian *body* ditandai dengan *tag* <body> di awal dan diakhiri dengan </body>. Bagian ini merupakan elemen terbesar dalam dokumen HTML yang berisi isi dokumen yang akan ditampilkan pada halaman web.

2.5.2 Cascading Style Sheet (CSS)

Cascading Style Sheet digunakan untuk mengatur *format* tata letak halaman web. CSS dapat digunakan untuk menentukan variasi teks, ukuran tabel, dan aspek-aspek lain dari halaman web yang sebelumnya hanya dapat didefinisikan dalam halaman HTML. Menurut Pfaffenberger, et al. (2004), CSS memungkinkan pengembang Web untuk menentukan informasi yang akan dipresentasikan tanpa mengganggu struktur informasinya. Informasi yang harus diketahui *browser* untuk memformat teks sebelumnya disimpan secara terpisah, dalam lembar gaya atau *Style Sheets*.

Lembar gaya mencantumkan gaya presentasi yang harus digunakan *browser* untuk menampilkan berbagai komponen dokumen, seperti judul, daftar, dan paragraf. Lembar gaya disimpan secara terpisah dari teks HTML. CSS membantu pengembang Web membuat tampilan yang seragam di beberapa halaman situs web. *Style* yang umum digunakan dalam situs web hanya perlu didefinisikan sekali dalam sebuah dokumen lembar gaya. Setelah *style* didefinisikan, *style* tersebut dapat digunakan oleh halaman manapun yang memiliki referensi dokumen lembar gaya.

2.5.3 Bootstrap

Bootstrap adalah *framework front-end* yang bersifat *open* untuk merancang halaman web. *Bootstrap* dapat membantu pengembang membuat tampilan antarmuka web menjadi lebih menarik, responsif, mudah dan dapat digunakan secara gratis (Alatas, 2013). *Bootstrap* menyediakan HTML, JSS, dan JavaScript untuk membuat komponen-komponen untuk membuat tampilan yang dinamis pada sebuah web.

2.5.4 Codeigniter

Codeigniter (CI) adalah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis (Sidik, 2012). *Codeigniter* adalah *framework* PHP yang membantu dalam pembuatan situs web dan aplikasi web untuk berbagai ukuran, tingkat dan kompleksitas, mulai dari situs web statis kecil hingga ke situs web tingkat perusahaan yang rumit.

2.5.5 PHP

Bahasa pemrograman PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang bekerja pada sebuah *web server*. Skrip-skrip PHP yang dibuat disimpan dalam sebuah *server* dan dieksekusi atau diproses dalam *server* tersebut. PHP memungkinkan sebuah situs web menjadi lebih interaktif dan dinamis (Madcoms, 2008). Menurut Brooks (2008), dokumen PHP dapat menampilkan keluaran pada layar *browser*, secara dasar dengan membuat halaman web berformat HTML sebagai keluaran tetapi tidak mengembalikan informasi yang dapat digunakan dalam dokumen HTML atau dengan skrip JavaScript.

Program PHP dapat dimulai dengan menuliskan *tag* pengenalan yang digunakan untuk menuliskan kode PHP, yaitu *tag* “<?php”, setelah menulis *tag* tersebut kemudian dapat dilanjutkan dengan isi kode program didalamnya. Untuk mengakhiri kode program yang dibuat, dapat ditutup dengan *tag* “?>”.

2.5.6 JavaScript

JavaScript adalah bahasa pemrograman berorientasi objek yang diinterpretasikan (bukan dikompilasi) dengan root pada C/C++ , yang dikembangkan sebagai alat untuk pembuatan Web (Brooks, 2007). *JavaScript* tidak beroperasi sebagai bahasa yang berdiri sendiri, melainkan dirancang untuk bekerja sama dengan HTML untuk membuat halaman Web menjadi interaktif. *JavaScript* tidak sama dengan Java, yang merupakan bahasa pemrograman berorientasi objek yang dikompilasi.

Javascript adalah bahasa yang case sensitive. Bahasa ini dapat membedakan penamaan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil. Sebagai contoh variabel dengan nama “test” berbeda dengan variabel dengan nama “TEST”. Setiap instruksi pada Javascript diakhiri dengan karakter titik koma (;).

2.6 Basis Data

Basis data adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Fathansyah, 2015). Kelompok data yang saling berhubungan ini disimpan secara bersama sedemikian rupa tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Basis data dikelola secara elektronik dengan bantuan komputer. Pada basis data terdapat pengaturan, pemilahan, pengelompokan, dan penngorganisasian data yang disimpan sesuai fungsi/jenisnya.

Pada basis data terdapat operasi-operasi dasar, meliputi : pembuatan basis data baru (*create database*), penghapusan basis data (*drop database*), pembuatan tabel baru pada suatu basis data (*create table*), penghapusan tabel dari suatu basis data (*drop table*), penambahan data baru (*insert*), pengambilan data dari sebuah tabel (*query*), pengubahan data dari sebuah tabel (*update*), dan penghapusan data dari sebuah tabel (*delete*).

2.6.1 Sistem Manajemen Basis Data

Menurut Kahate (2004), sistem manajemen basis data adalah seperangkat program yang digunakan untuk menyimpan, memperbaharui, dan mengambil basis data. Sistem manajemen basis data juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pengaturan konsistensi data, dan sebagainya. Perangkat lunak yang termasuk sistem manajemen basis data diantaranya : dBase, FoxBase, Rbase, Microsoft-Access, MS-SQL Server, Oracle Database, IBM DB2, Informix, Sybase, MySQL dan PostgreSQL.

2.6.2 MySQL

Menurut Peicevic (2017), MySQL adalah salah satu sistem manajemen basis data relasional (*Relational Database Management System* atau disingkat RDBMS) yang paling populer. MySQL bersifat *open source* dan tersedia di bawah lisensi General Public License (GNU). MySQL dapat dijalankan, dibagi dan bahkan dimodifikasi secara gratis. MySQL banyak digunakan sebagai RDBMS untuk pengembangan situs Web karena cepat dan mudah digunakan.

MySQL menggunakan *format* standar dari bahasa data yang populer yaitu SQL. SQL merupakan singkatan dari Structures Query Language yaitu bahasa pemrograman dengan tujuan khusus yang dirancang untuk mengelola data yang disimpan pada sistem manajemen basis data relasional. Contoh dari syntax untuk mengambil data dari tabel Customer yang tinggal di kota Berlin adalah sebagai berikut :

```
SELECT name FROM customers WHERE city =  
'Berlin';
```

2.7 Pengujian Perangkat Lunak

Pengujian adalah serangkaian kegiatan yang direncanakan dan dilakukan secara sistematis (Pressman, 2010). Pada perangkat lunak, pengujian dilakukan untuk menemukan kesalahan yang tidak sesuai dengan rancangan sistem yang telah dibuat. Pengujian dimulai pada komponen yang kecil ke komponen yang lebih besar hingga menuju integrasi sistem secara keseluruhan.

Terdapat dua jenis pengujian pada pengujian perangkat lunak yaitu *White Box Testing* atau pengujian kotak putih dan *Black Box Testing* atau pengujian kotak hitam. Pengujian kotak putih (*White Box Testing*) adalah jenis pengujian yang dilakukan ketika tim penguji memiliki kode sumber (*source code*) dan kode yang dapat dieksekusi (*executable code*). Pengembang dan tim penguji memiliki kesempatan untuk meninjau dan menguji setiap baris kode itu (McLeod, 2007). Pengujian ini fokus untuk menguji keamanan, arus masukan dan keluaran yang ada pada sistem, dan meningkatkan perancangan dan kegunaan perangkat lunak. Pengujian dilakukan pada tingkat yang berbeda yang melibatkan sistem lengkap atau bagian-bagiannya sepanjang siklus hidup suatu produk perangkat lunak.

Menurut McLeod (2007), jenis pengujian kotak hitam dapat dilakukan ketika tim penguji tidak memiliki kode sumber melainkan hanya kode yang dapat dieksekusi. Pengujian kotak hitam juga dikenal sebagai pengujian perilaku sistem

karena dominasi analisis hasil yang diharapkan untuk teknik ini. Pengujian perangkat lunak dilakukan pada tiga level yaitu : pengujian unit untuk , pengujian integrasi, dan pengujian validasi untuk mengakomodasi tingkat pengujian mulai dari tingkat terkecil hingga terbesar.

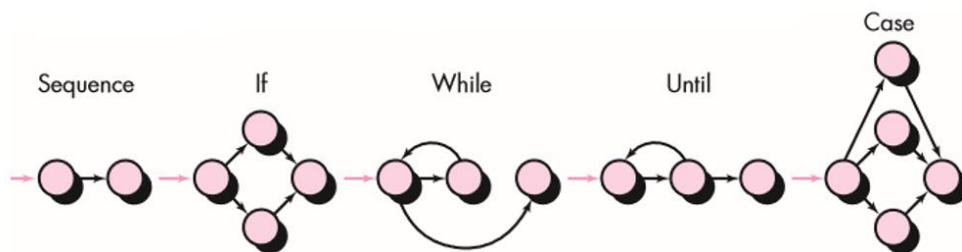
2.7.1 Pengujian Unit

Pengujian unit berfokus pada struktur data dan logika pemrosesan dalam batas-batas komponen. Pengujian unit dilakukan dengan menggunakan jenis kotak putih dengan teknik *basis-path testing*. Teknik pengujian *basis-path* memungkinkan penguji untuk mengetahui ukuran kompleksitas sistem dan menggunakan ukuran tersebut sebagai acuan untuk mendefinisikan jalur-jalur eksekusi (Pressman, 2010).

Tahap *basis path testing* menurut Pressman (2010), yaitu :

1. Mendefinisikan notasi *flow graph* berdasarkan kode program sebagai dasar.

Notasi *flow graph* dapat dilihat pada gambar 2.6.



Gambar 2.6 Notasi Flow Graph

Sumber : Pressman (2010)

Berdasarkan gambar 2.6, setiap lingkaran disebut sebagai *flow graph node* yang merepresentasikan satu atau lebih pernyataan prosedural. Panah pada *flow graph* disebut *edge* (tepi) atau penghubung (*link*) yang merepresentasikan aliran kontrol. Area yang dibatasi oleh *node* dan *edge* disebut *region*.

2. Menentukan *cyclomatic complexity* berdasarkan hasil *flow graph*

Setelah mendapatkan jalur independen pada *flowgraph*, untuk menentukan berapa banyak jalur yang dicari, dilakukan komputasi *cyclomatic complexity* yang menyediakan ukuran kuantitatif dari kompleksitas logika dari sebuah program.

Cyclomatic complexity $V(G)$ dari sebuah *flow graph* G didefinisikan sebagai :

$V(G) = E - N + 2$ dimana E adalah jumlah *edge* dan N adalah jumlah *node*

$V(G) = P + 1$ dimana P adalah jumlah *predicate node*

$V(G) = R$ dimana R adalah jumlah *region*

3. Menentukan jalur independen

Setelah mendapatkan nilai $V(G)$, maka nilai tersebut adalah jumlah jalur independen yang dapat dihasilkan dari *flow graph*.

4. Menyiapkan kasus uji dengan menjalankan masing-masing jalur independen yang telah ditentukan.

2.7.2 Pengujian Integrasi

Pengujian integrasi adalah teknik sistematis untuk membangun arsitektur perangkat lunak sementara pada saat bersamaan melakukan pengujian untuk menemukan kesalahan yang terkait dengan antarmuka (Pressman, 2010). Tujuan pengujian ini adalah untuk mengambil komponen pengujian unit dan membangun sebuah struktur program yang telah ditentukan pada perancangan.

2.7.3 Pengujian Validasi

Pengujian adalah jenis pengujian kotak hitam dimana penguji tidak mengetahui kode program namun hanya menjalankan fungsionalitas sistem. Menurut Pressman (2010), pengujian validasi dimulai pada puncak dari pengujian integrasi, pada saat komponen individu telah diuji dan perangkat lunak telah dirangkai sebagai sebuah paket dan kesalahan pada antarmuka telah ditemukan dan diperbaiki. Pengujian validasi berfokus pada aksi pengguna terhadap sistem dan *output* yang dihasilkan sistem.

Penulis menggunakan pengujian validasi dengan metode pengujian berdasarkan skenario atau *scenario-based testing* yang fokus pada apa yang pengguna lakukan pada sistem, bukan produk apa yang dihasilkan oleh sistem. Metode ini menggunakan *task* pada *use case* yang harus dijalankan oleh pengguna dan mengaplikasikannya sebagai sebuah pengujian (Pressman, 2010).

2.7.4 Pengujian Compatibility

Pengujian *compatibility* termasuk kedalam jenis pengujian non-fungsional. Pengujian ini memeriksa apakah perangkat lunak yang dibuat dapat berjalan pada berbagai perangkat keras, sistem operasi, aplikasi, jaringan, atau perangkat *mobile* (Guru99, 2018). Pada penelitian ini, digunakan *tool* yaitu SortSite untuk menguji *compatibility* perangkat lunak yang dibangun.

SortSite adalah alat untuk menguji kualitas sebuah situs web. Masalah pada kualitas sebuah web meliputi *compatibility*, *usability*, adanya *link* yang rusak, memastikan web yang dibuat tidak melanggar hukum, dan mengikuti standar dalam pembuatan web. SortSite telah banyak digunakan oleh banyak perusahaan dan konsultan independen. Cara kerjanya yaitu dengan memindai setiap halaman dari sebuah halaman situs web dan memeriksa *link* halaman tersebut dengan halaman lain yang belum dipindai. Proses pemindaian ini terus berjalan sampai semua halaman web telah diperiksa (PowerMapper, 2015).

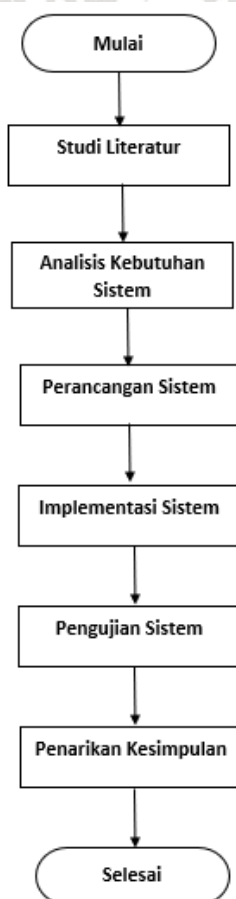
BAB 3 METODOLOGI PENELITIAN

3.1 Tipe Penelitian

Berdasarkan produk yang akan dihasilkan, penelitian ini bertipe implementatif – pengembangan. Penelitian implementatif akan menghasilkan solusi terhadap permasalahan yang diangkat berupa produk yang dalam penelitian ini berupa produk perangkat lunak. Pengembangan adalah salah satu jenis penelitian implementatif untuk membuat suatu produk perangkat lunak utuh yang sebelumnya belum ada menjadi ada dengan menggunakan prinsip rekayasa pengembangan perangkat lunak yang didalamnya terdapat analisis kebutuhan, perancangan, implementasi, dan pengujian. Pada penelitian ini akan menghasilkan produk perangkat lunak yaitu sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* dengan mengangkat permasalahan yang ada pada Klinik Medis Elisa Malang.

3.2 Strategi dan Rancangan Penelitian

Pada penelitian ini terdapat beberapa tahap dalam proses perancangan dan implementasi sistem. Langkah-langkah tersebut terdapat pada diagram alur metodologi penelitian berikut :



Gambar 3.1 Tahapan Metodologi Penelitian

3.2.1 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk memperoleh teori dan informasi yang berkaitan dengan penelitian yang dilakukan. Studi literatur pada penelitian ini menggunakan buku dan jurnal yang berkaitan dengan :

1. Rekam medis dan rekam medis elektronik
2. *Software Development Life Cycle : Waterfall Model*
3. Teknologi Pengembangan Sistem
4. Basis Data dan Sistem Manajemen Basis Data
5. Unified Modelling Language (UML)
6. Pengujian Perangkat Lunak

3.2.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan mendefinisikan dan menentukan kebutuhan yang diperlukan dalam merancang perangkat lunak. Kebutuhan yang telah diidentifikasi kemudian dimodelkan dalam diagram *use case* untuk mendefinisikan kebutuhan fungsional sistem dan *use case scenario* yaitu untuk menjabarkan setiap fungsional sistem. Analisis Kebutuhan dilakukan dengan cara wawancara dan observasi langsung ke Klinik Medis Elisa Malang.

3.2.3 Perancangan Sistem

Perancangan sistem dilakukan setelah analisis kebutuhan sistem selesai. Pada perancangan sistem, penulis menggunakan pendekatan berorientasi objek dan menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Hasil dari tahap perancangan adalah *sequence diagram* yang menjelaskan alur pertukaran pesan antarobjek terhadap garis waktu dan *class diagram* yang menunjukkan kelas-kelas apa saja yang terdapat pada sistem dan hubungan antarkelas. Pada perancangan sistem, dilakukan pula perancangan antarmuka, perancangan algoritme, dan perancangan basis data.

3.2.4 Implementasi

Implementasi dilakukan dengan menggunakan hasil analisis kebutuhan dan perancangan sebagai acuan. Sistem ini dibangun dengan menggunakan bahasa dan teknologi berikut :

1. Bahasa markah HTML 5
2. Bahasa lembar gaya CSS 3 dengan kerangka kerja Bootstrap 3
3. Bahasa skrip PHP 5 dengan kerangka kerja Codeigniter 3
4. Bahasa skrip Javascript dengan kerangka kerja jQuery 3
5. Basis Data MySQL

3.2.5 Pengujian Sistem

Pengujian sistem dilakukan untuk memeriksa apakah hasil perangkat lunak sudah sesuai dengan analisis kebutuhan dan perancangan. Pengujian dilakukan pada level unit dan validasi. Pada sistem ini, digunakan jenis pengujian *Whitebox Testing* untuk menguji apakah kode program dan logika yang terdapat didalamnya dapat dijalankan dengan benar dan *Blackbox Testing* untuk menguji fungsionalitas sistem.

3.2.6 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan setelah tahap pengujian selesai. Kesimpulan disesuaikan dengan rumusan masalah yang ada. Kemudian penulis memberikan saran untuk memudahkan penelitian lebih lanjut tentang topik terkait.



BAB 4 REKAYASA KEBUTUHAN

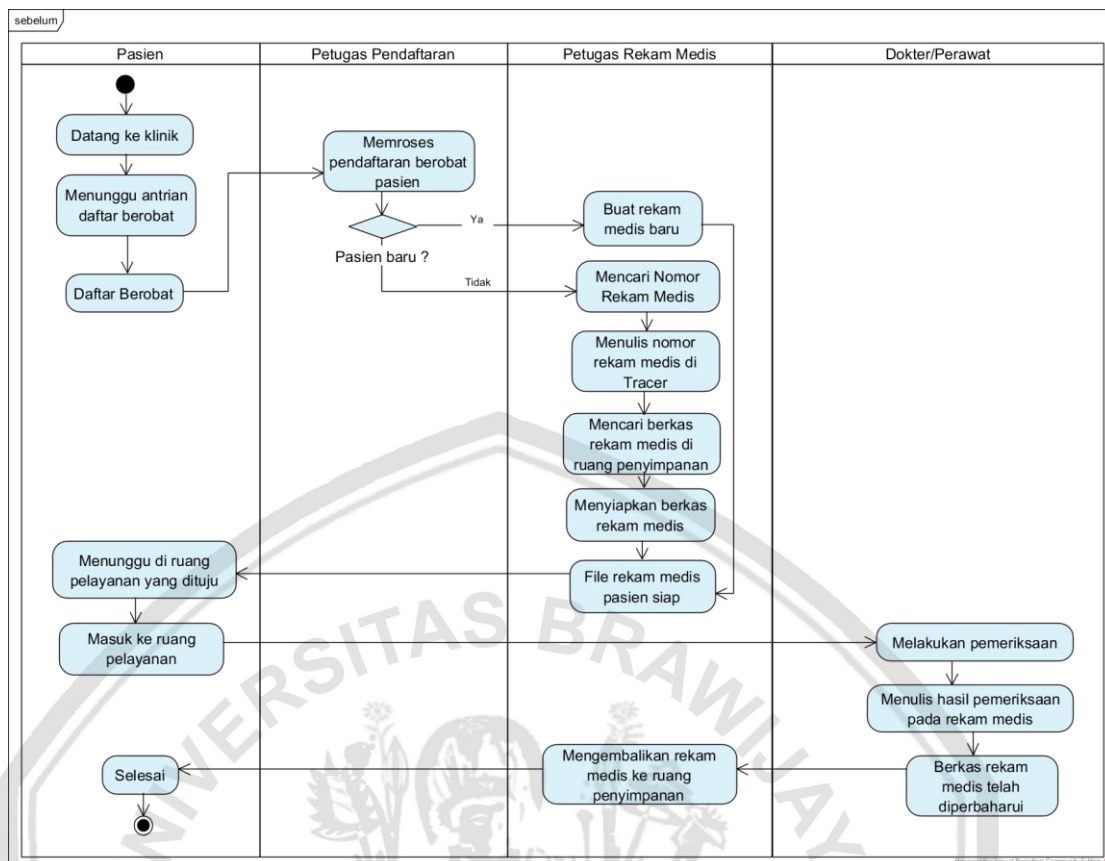
4.1 Elisitasi Kebutuhan

Elisitasi kebutuhan dilakukan dengan cara observasi langsung dan wawancara mengenai sistem pendaftaran, pengambilan dan pengembalian rekam medis di Klinik Medis Elisa. Wawancara dilakukan dengan narasumber perawat, petugas pendaftaran, dan manajer operasional Klinik Medis Elisa. Penulis melakukan wawancara dengan tujuan untuk memperoleh informasi mengenai isi rekam medis yang digunakan dan proses pelaporan pasien rawat jalan ke Dinas Kesehatan Kota Malang.

Dari hasil wawancara, didapatkan hasil berupa format laporan pasien rawat jalan ke Dinas Kesehatan Kota Malang yang terdiri dari jumlah kunjungan rawat jalan dan 10 diagnosa terbanyak (Lampiran A) serta satu bendel *file* rekam medis yang didalamnya berisi :

1. Formulir Identitas Pasien Baru untuk mengisi data personal pasien yang belum mempunyai rekam medis atau baru pertama kali berobat ke Klinik Medis Elisa (Lampiran B).
2. Formulir Pengkajian Klinis Medis Rawat Jalan yang digunakan untuk menuliskan anamnesa, pemeriksaan fisik pasien, pemeriksaan penunjang pasien, faktor resiko, anjuran dokter, diagnosa, dan rencana pemecahan masalah (Lampiran C).
3. Formulir Rekam Medis Umum untuk yang didalamnya terdapat kolom keluhan dan pemeriksaan, diagnosa, terapi, keterangan, dan kode ICD (Lampiran D).
4. Formulir Rekam Medis Poli Gigi yang berisi odontogram, kolom keluhan dan pemeriksaan, diagnosa, terapi, keterangan, dan kode ICD (Lampiran E).
5. Formulir Pemberian Informasi dan Edukasi Rawat Jalan untuk mengisi materi edukasi rawat jalan yang dijelaskan kepada pasien (Lampiran F).
6. Formulir Permintaan Pemeriksaan Diagnostik untuk mengisi pemeriksaan diagnostik apa saja yang akan diterima pasien (Lampiran G).

Alur pelayanan pasien dari mulai pendaftaran sampai pemeriksaan pasien digambarkan dengan *activity diagram* yang digunakan untuk merepresentasikan aliran dari aktivitas satu ke aktivitas lainnya. *Activity diagram* untuk alur pelayanan pasien saat ini dijelaskan pada Gambar 4.1.



Gambar 4.1 Alur Pelayanan Pasien Klinik Medis Elisa Saat Ini

Penjelasan alur pelayanan pasien rawat jalan Klinik Medis Elisa yaitu pertama, pasien datang ke klinik dan mengantri untuk pendaftaran berobat rawat jalan. Kemudian pasien melakukan pendaftaran ke loket. Pasien akan menulis nama, nomor kartu (untuk pasien asuransi), dan keperluan di buku pendaftaran. Kemudian petugas pendaftaran akan mencari nomor rekam medis pasien dengan mengetikkan nama pasien pada komputer. Untuk pasien yang baru pertama berobat, petugas akan memberikan *formulir* pendataan pasien baru dan pasien akan mengisinya kemudian menyerahkan kembali ke petugas. Petugas akan menginisialisasi nomor rekam medis baru tersebut dan menyiapkan satu bendel *file* rekam medis baru. Langkah selanjutnya yaitu petugas mencari nomor rekam medis pasien pada komputer, kemudian petugas menulis nomor rekam medis pasien pada *tracer*. *Tracer* tersebut berupa kertas yang berisi nomor rekam medis pasien yang akan diambil. Petugas rekam medis mencari rekam medis sesuai dengan *tracer* yang diberikan dan menyiapkan berkas rekam medis untuk diberikan ke pelayanan yang dituju pasien.

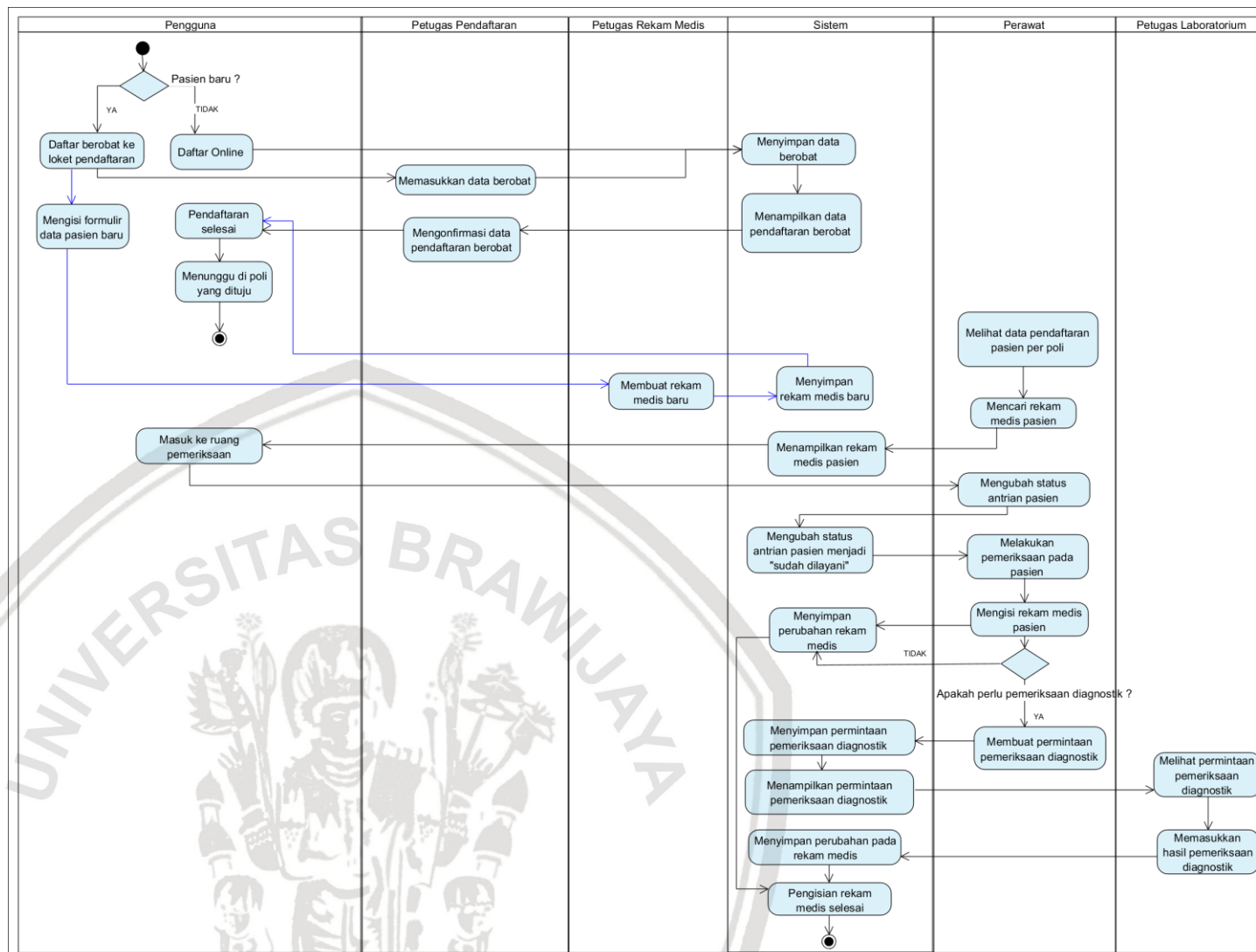
Selama pasien diperiksa, perawat atau dokter akan menulis hasil pemeriksaan ke dalam berkas rekam medis. Setelah pasien selesai diperiksa, berkas rekam medis diberikan kembali ke petugas rekam medis untuk disimpan di ruang penyimpanan.

4.2 Deskripsi Sistem

Sistem yang akan dibangun adalah sistem yang menyediakan pendaftaran berobat (*control* berobat) pasien yang dapat diakses secara *online* yang bertujuan menghemat waktu pasien dan manajemen rekam medis elektronik untuk mengelola rekam medis pasien rawat jalan sehingga mempermudah penyimpanan rekam medis dan lebih menghemat waktu dalam hal pencarian rekam medis. Sistem ini dibuat dengan berbasis web.

Sesuai dengan hasil wawancara dan observasi, aktor pada sistem ini adalah *guest* yang dapat mengakses fungsi pendaftaran untuk berobat dan dapat *log in* untuk mengakses fungsi utama sistem. *Log in* hanya dapat dilakukan oleh pengguna yang berhubungan dengan manajemen rekam medis karena tidak tersedia fitur registrasi untuk dapat masuk ke dalam sistem rekam medis. Lalu terdapat aktor petugas pendaftaran yang dapat mengakses fungsi yang berhubungan dengan pendaftaran berobat pasien rawat jalan. Petugas rekam medis yang dapat melakukan pencarian dan pembuatan rekam medis. Perawat yang dapat mengisi rekam medis pasien sesuai yang tertulis pada rekam medis asli yang telah diisi oleh dokter. Kemudian ada petugas laboratorium yang mengakses fungsi-fungsi pemeriksaan diagnostik. Alur sistem yang ditawarkan digambarkan ke dalam *activity diagram* pada Gambar 4.2





Gambar 4.2 Alur Sistem yang Ditawarkan

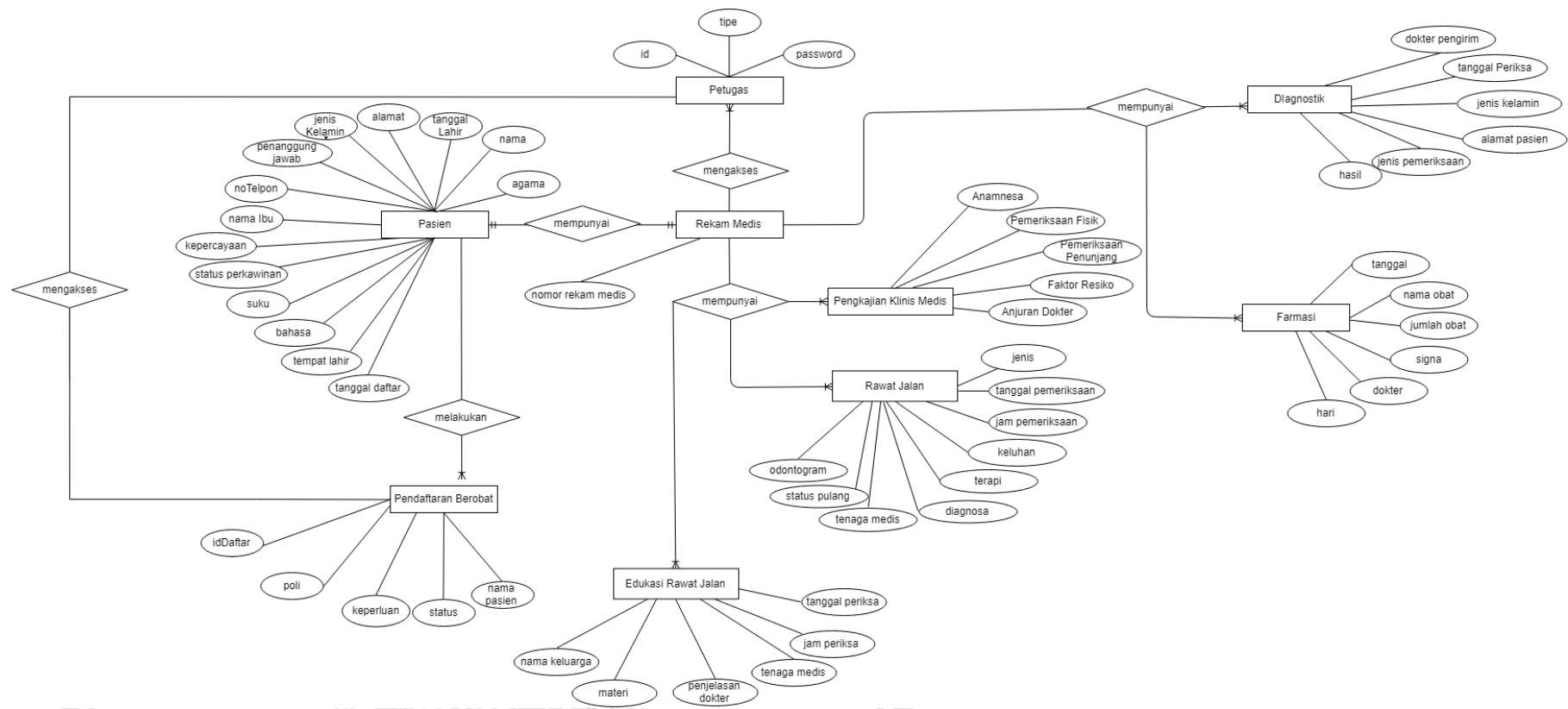
Penjelasan alur dari sistem yang akan dibangun yaitu pertama pasien yang sudah pernah berobat di Klinik Elisa melakukan pendaftaran berobat pada web pendaftaran, sedangkan untuk pasien baru diharuskan datang ke klinik untuk mengisi data pendaftaran pasien baru. Kemudian pasien datang ke klinik untuk konfirmasi kedatangan ke petugas pendaftaran. Setelah itu petugas rekam medis akan mencari rekam medis pasien pada sistem, jika tidak ditemukan maka akan membuat rekam medis baru. Setelah data rekam medis siap, pasien menunggu di ruang pelayanan yang dituju. Setelah tiba giliran pasien masuk ke ruang pemeriksaan, perawat akan mengubah status pada antrian pasien menjadi “sudah dilayani”.

Setelah pasien masuk ke ruang pemeriksaan, perawat atau dokter melakukan pemeriksaan pada pasien kemudian mengisi data kunjungan pasien pada rekam medis dan mencatat terapi yang didapat pasien ke dalam rekam medis. *File* rekam medis terdiri dari data pribadi pasien, data pengkajian klinis medis rawat jalan, data kesehatan umum, data kesehatan mulut dan gigi, data edukasi rawat jalan, dan data riwayat medis. Jika pasien memerlukan pemeriksaan diagnostik, perawat akan membuat permintaan pemeriksaan laboratorium untuk dikirim ke bagian laboratorium. Kemudian hasil dari pemeriksaan laboratorium pasien akan disimpan kedalam rekam medis pasien.

Sistem ini dibuat dengan menggunakan bahasa pemrograman PHP dan juga menggunakan *framework bootstrap* dan *codeigniter*. Sistem manajemen basis data yang digunakan adalah MySQL. Sistem ini dapat dijalankan pada berbagai *browser* seperti Google Chrome, Mozilla Firefox, Opera, dan Microsoft Edge.

4.3 Analisis Data

Analisis data dilakukan untuk menemukan entitas apa saja yang terlibat dalam sistem rekam medis elektronik dan pendaftaran pasien *online* di Klinik Medis Elisa. Hasil dari analisis data digambarkan pada *entitiy relationship diagram* yang memuat entitas-entitas dan hubungan antarentitas pada sistem. Pada sistem pendaftaran pasien dan rekam medis elektronik studi kasus Klinik Medis Elisa terdapat entitas pasien, petugas, pendaftaran berobat, dan rekam medis yang terdiri dari beberapa bagian yaitu rawat jalan, pengkajian klinis medis, edukasi rawat jalan, diagnostik, dan farmasi. *Entity Relationship Diagram* ditunjukkan pada Gambar 4.3.



Gambar 4.3 Entity Relationship Diagram

4.4 Identifikasi Pengguna

Berdasarkan hasil elisitasi kebutuhan, diperoleh enam pengguna sistem yaitu *guest*, petugas pendaftaran, petugas rekam medis, perawat, dan petugas laboratorium yang dideskripsikan pada Tabel 4.1.

Tabel 4.1 Identifikasi Pengguna

No	Klasifikasi Pengguna	Deskripsi Pengguna
1.	<i>Guest</i>	<i>Guest</i> dapat melakukan pendaftaran berobat yang dilakukan dua jam sebelum datang ke klinik. Pendaftaran berobat ini khusus untuk pasien yang sudah pernah berobat ke klinik. <i>Guest</i> juga dapat <i>log in</i> ke sistem sebagai salah satu pengguna dengan fungsinya masing-masing.
2.	Petugas Pendaftaran	Petugas pendaftaran memiliki fungsi yang terbatas pada pendaftaran pasien yaitu dapat melakukan pendaftaran pasien dan melihat data pendaftaran pasien.
3.	Petugas Rekam Medis	Petugas Rekam Medis memiliki fungsi yang berhubungan dengan rekam medis pasien, yaitu melakukan pencarian rekam medis dan pembuatan rekam medis baru.
4.	Perawat	Perawat dapat melihat antrian pasien umum, gigi, dan kesehatan ibu dan anak sesuai dengan menu yang dipilih sebelumnya dan dapat melakukan pengisian rekam medis.
5.	Petugas Laboratorium	Petugas laboratorium memiliki fungsi yang terbatas pada pemeriksaan diagnostik yaitu melihat daftar blanko pemeriksaan diagnostik, melihat blanko pemeriksaan diagnostik, dan menambahkan hasil pemeriksaan diagnostik ke rekam medis.

4.5 Identifikasi Kebutuhan

Identifikasi kebutuhan terdiri dari identifikasi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional didefinisikan setelah melakukan identifikasi pengguna dan wawancara mengenai kebutuhan apa yang harus ada dalam sistem. Kebutuhan fungsional adalah layanan yang disediakan sistem yang akan dioperasikan oleh pengguna sedangkan kebutuhan non-fungsional adalah batasan layanan dari sistem.

4.5.1 Kebutuhan Fungsional

Setiap kebutuhan fungsional sistem diberi kode dengan penamaan RME_F_XXX. RME merupakan singkatan dari nama sistem yang dibuat yaitu Rekam Medis Elisa, F merupakan tanda bahwa kebutuhan tersebut adalah kebutuhan fungsional, dan XXX merupakan nomor dari kebutuhan fungsional. Kebutuhan fungsional pada sistem yang dibangun ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsional Sistem

No.	Kode	Kebutuhan	Nama Use Case
1	RME_F_001	Sistem menyimpan data pendaftaran berobat bagi pasien yang sudah pernah berobat ke Klinik Elisa dengan mengakses halaman web.	Menyimpan Pendaftaran Berobat
1.1	RME_F_001_01	Sistem menampilkan <i>formulir</i> pendaftaran yang harus diisi pengguna dan tombol “simpan”.	
1.2	RME_F_001_02	Data yang diisi yaitu nomor member pasien, pilihan poli, keperluan, dan jam datang.	
2	RME_F_002	Sistem memulai <i>session</i> untuk pengguna <i>guest</i> , sehingga pengguna dapat mengakses fungsionalitas yang ada pada sistem berdasarkan jenis pengguna.	Log in
2.1	RME_F_002_02	Sistem menampilkan halaman <i>log in</i> berisi <i>field username</i> dan <i>password</i> yang harus diisi pengguna dan tombol <i>log in</i> untuk dapat masuk ke sistem.	
3	RME_F_003	Sistem dapat mengakhiri <i>session</i> yang sedang berlangsung.	Log out
4	RME_F_004	Sistem menampilkan data pasien yang telah mendaftar untuk berobat.	Melihat list pendaftaran berobat
4.1	RME_F_004_01	List pendaftaran pasien yang ditampilkan berbentuk tabel yang berisi ID pendaftaran, nomor member, tanggal daftar, nama pasien, umur, poli, keperluan, nomor telpon, sumber, dan status.	
5	RME_F_005	Sistem menampilkan jumlah kunjungan pasien berdasarkan bulan dan poli.	Melihat jumlah kunjungan rawat jalan
5.1	RME_F_005_01	Sistem menampilkan halaman yang berisi pilihan poli, bulan, dan tahun kunjungan rawat jalan.	
6	RME_F_006	Sistem mencetak halaman yang berisi jumlah kunjungan rawat jalan yang sedang ditampilkan.	Cetak jumlah kunjungan rawat jalan
6.1	RME_F_006_01	Sistem menampilkan <i>print lay out</i> setelah pengguna memilih menu “cetak”.	

7	RME_F_007	Sistem menampilkan diagnosa terbanyak pasien rawat jalan berdasarkan bulan dan poli yang dipilih.	Melihat Peringkat Diagnosa
7.1	RME_F_007_01	Sistem menampilkan halaman yang berisi pilihan poli bulan dan tahun.	
8	RME_F_008	Sistem mencetak halaman yang berisi diagnosa terbanyak yang sedang ditampilkan.	Cetak peringkat diagnosa
8.1	RME_F_008_01	Sistem menampilkan <i>print lay out</i> setelah pengguna memilih menu "cetak".	
9	RME_009	Sistem melakukan pengubahan status konfirmasi pendaftaran pasien pada tabel pendaftaran berobat.	Mengubah status konfirmasi pendaftaran berobat
10	RME_F_010	Sistem menyimpan data rekam medis baru.	Membuat rekam medis baru
10.1	RME_F_010_01	Sistem menampilkan <i>formulir</i> rekam medis baru dan tombol untuk menyimpan.	
11	RME_F_011	Sistem menampilkan data pribadi pasien yang telah tersimpan.	Melihat data pribadi pasien
12	RME_F_012	Sistem menyimpan pengubahan data pribadi pasien pada rekam medis.	Mengubah data pribadi pasien
12.1	RME_F_012_01	Sistem menampilkan halaman yang berisi <i>form</i> data pribadi pasien dan tombol untuk menyimpan perubahan.	
13	RME_F_013	Sistem merekam penghapusan rekam medis pasien.	Menghapus rekam medis
14	RME_F_014	Sistem menampilkan <i>list</i> pendaftaran pasien berdasarkan poli yang dipilih.	Melihat <i>list</i> pendaftaran pasien per poli
14.1	RME_F_014_01	Sistem menampilkan menu "Pelayanan Umum", "Pelayanan Gigi", dan "Kesehatan Ibu dan Anak".	
14.2	RME_F_014_02	Sistem menampilkan <i>list</i> pendaftaran pasien berbentuk tabel yang berisi ID pendaftaran, tanggal daftar, nama pasien, umur, poli, keperluan, nomor telpon, sumber, dan status.	

15	RME_F_015	Sistem menyimpan data perubahan status antrian pasien yang awalnya “belum dilayani” menjadi “sudah dilayani”.	Mengubah status antrian pasien
16	RME_F_016	Sistem menampilkan isi <i>file</i> rekam medis pasien.	Melihat Rekam Medis Pasien
16.1	RME_F_016_01	Sistem melakukan pencarian rekam medis sebelum dapat melihat isi rekam medis pasien.	
16.2	RME_F_016_02	Sistem menampilkan halaman pencarian rekam medis yang berisi <i>field</i> untuk mengetikkan nama pasien dan tombol “Cari”.	
16.3	RME_F_016_03	Sistem menampilkan hasil pencarian rekam medis dalam bentuk tabel.	
16.4	RME_F_016_04	Sistem menampilkan data rekam medis pasien terdiri dari beberapa <i>tab</i> yaitu data pribadi pasien, pengkajian klinis medis, kesehatan umum, kesehatan gigi dan mulut, edukasi rawat jalan, dan riwayat medis.	
17	RME_F_017	Sistem menampilkan riwayat pengkajian klinis medis pasien.	Melihat riwayat pengkajian klinis medis.
17.1	RME_F_017_01	Sistem menampilkan yang berisi data tanggal pemeriksaan, keluhan, dan petugas. Data ditampilkan dalam bentuk tabel.	
18	RME_F_018	Sistem menampilkan <i>detail</i> data pengkajian klinis medis rawat jalan pasien yang telah dipilih.	Melihat <i>detail</i> data pengkajian klinis medis rawat jalan
18.1	RME_F_018_01	Sistem menampilkan detail pengkajian klinis medis berupa anamnesa, pemeriksaan fisik pasien, pemeriksaan penunjang pasien, faktor resiko, anjuran dokter, diagnosa, dan rencana pemecahan masalah.	
19	RME_F_019	Sistem menyimpan pembuatan pengkajian klinis medis baru.	Membuat pengkajian klinis medis rawat jalan baru
19.1	RME_F_019_01	Sistem menampilkan <i>formulir</i> pengkajian klinis medis baru dan tombol untuk menyimpan.	
19.2	RME_F_019_02	Data yang diisikan pengguna pada formulir yaitu anamnesa, pemeriksaan fisik pasien, pemeriksaan penunjang pasien, faktor resiko, anjuran dokter,	

		diagnosa, dan rencana pemecahan masalah.	
20	RME_F_020	Sistem menyimpan penambahan data rawat jalan umum untuk pasien baru.	Membuat data rawat jalan umum baru
20.1	RME_F_020_01	Sistem menampilkan <i>form</i> yang berisi keluhan, terapi, dan diagnosa yang harus diisi pengguna.	
21	RME_F_021	Sistem menyimpan penambahan data rawat jalan gigi untuk pasien baru.	Membuat data rawat jalan gigi baru
21.1	RME_F_021_01	Sistem menampilkan <i>form</i> yang berisi keluhan, terapi, dan diagnosa yang harus diisi pengguna.	
22	RME_F_022	Sistem menampilkan <i>detail</i> data rawat jalan pelayanan umum.	Melihat <i>detail</i> data rawat jalan umum
22.1	RME_F_022_01	Sistem menampilkan data rawat jalan pasien berdasarkan data yang dipilih pada tabel riwayat kunjungan poli umum.	
23	RME_F_023	Sistem menampilkan <i>detail</i> data rawat jalan pelayanan gigi	Melihat <i>detail</i> data rawat jalan gigi
23.1	RME_F_023_01	Sistem menampilkan data rawat jalan berdasarkan data yang dipilih pada tabel riwayat kunjungan poli gigi.	
24	RME_F_024	Sistem menyimpan penambahan data rawat jalan untuk pasien umum.	Menambah data rawat jalan umum
24.1	RME_F_026_01	Sistem menampilkan <i>form</i> yang berisi keluhan, terapi, dan diagnosa yang akan ditampilkan setelah pengguna menekan tombol untuk menambah data rawat jalan.	
25	RME_F_025	Sistem menyimpan penambahan data rawat jalan untuk pasien gigi.	Menambah data rawat jalan gigi
25.1	RME_F_027_01	Sistem menampilkan <i>form</i> yang berisi keluhan, terapi, dan diagnosa yang akan ditampilkan setelah pengguna menekan tombol untuk menambah data rawat jalan.	
26	RME_F_026	Sistem menampilkan riwayat kunjungan pasien selama berobat di poli umum.	Melihat riwayat kunjungan umum
26.1	RME_F_026_01	Sistem menampilkan riwayat kunjungan pasien dalam bentuk	

		tabel yang berisi pelayanan, tanggal pelayanan, dan diagnosa.	
27	RME_F_027	Sistem menampilkan riwayat kunjungan pasien selama berobat di poli gigi.	Melihat riwayat kunjungan gigi
27.1	RME_F_027_01	Sistem menampilkan riwayat kunjungan pasien dalam bentuk tabel yang berisi pelayanan, tanggal pelayanan, dan diagnosa.	
28	RME_F_028	Sistem menampilkan riwayat pengobatan yang diterima pasien selama berobat di Klinik Elisa dalam bentuk tabel.	Melihat riwayat farmasi
29	RME_F_029	Sistem menyimpan penambahan data terapi farmasi	Menambah data terapi farmasi
29.1	RME_F_029_01	Sistem menampilkan <i>formulir</i> terapi farmasi dan tombol “simpan”.	
30	RME_F_030	Sistem menampilkan detail pemberian obat yang dipilih dari tabel riwayat farmasi pada rekam medis pasien.	Melihat detail terapi farmasi
31	RME_F_031	Sistem menampilkan riwayat diagnostik yang diterima pasien selama berobat di Klinik Elisa dalam bentuk tabel.	Melihat riwayat diagnostik
32	RME_F_032	Sistem menampilkan hasil pemeriksaan diagnostik pasien.	Melihat hasil pemeriksaan diagnostik
32.1	RME_F_032_01	Sistem menampilkan tabel riwayat diagnostik dan pengguna memilih hasil pemeriksaan diagnostik yang akan dilihat.	
33	RME_F_033	Sistem menyimpan penambahan data permintaan pemeriksaan diagnostik.	Menambah permintaan pemeriksaan diagnostik
33.1	RME_F_033_01	Sistem menampilkan formulir permintaan diagnostik yang berisi tanggal permintaan, nama pasien, umur, alamat, jenis kelamin, dokter pengirim, dan jenis pemeriksaan.	
34	RME_F_034	Sistem menampilkan riwayat pemberian informasi dan edukasi rawat jalan dalam bentuk tabel.	Melihat riwayat pemberian informasi dan edukasi rawat jalan
35	RME_F_035	Sistem menampilkan detail dari data pemberian informasi dan edukasi rawat jalan yang telah dipilih.	Melihat detail data pemberian informasi dan edukasi rawat jalan

35.1	RME_F_035_01	Sistem menampilkan halaman yang berisi materi edukasi dan penjelasan dokter kepada pasien.	
36	RME_F_036	Sistem menyimpan penambahan data pemberian informasi dan edukasi rawat jalan pada pasien.	Menambah data pemberian informasi dan edukasi rawat jalan
36.1	RME_F_036_01	Sistem menampilkan <i>form</i> pemberian informasi dan edukasi rawat jalan dan tombol untuk menyimpan.	
37	RME_F_037	Sistem menampilkan daftar permintaan pemeriksaan diagnostik yang telah diterima dalam bentuk tabel.	Melihat daftar permintaan pemeriksaan diagnostik
38	RME_F_038	Sistem menampilkan detail permintaan pemeriksaan diagnostik yang dipilih dari tabel permintaan pemeriksaan diagnostik atau tabel riwayat diagnostik pada rekam medis pasien.	Melihat detail permintaan pemeriksaan diagnostik
39	RME_F_039	Sistem menyimpan penambahan hasil pemeriksaan diagnostik pada rekam medis pasien.	Menambah hasil pemeriksaan diagnostik
39.1	RME_F_039_01	Sistem menampilkan <i>form</i> untuk menambahkan hasil pemeriksaan diagnostik dan tombol untuk menyimpan hasil pada rekam medis.	

Pemetaan *use case* dengan aktor pada sistem dijabarkan pada Tabel 4.3 berikut.

Tabel 4.3 Pemetaan Use Case

Kode	Nama Use Case	Aktor
RME_F_001	Menyimpan Pendaftaran Berobat	Guest, Petugas Pendaftaran
RME_F_002	<i>Log In</i>	Guest
RME_F_003	<i>Log Out</i>	Petugas Pendaftaran, Petugas Rekam Medis, Perawat, Petugas Laboratorium, Petugas Farmasi
RME_F_004	Melihat <i>List</i> Pendaftaran Berobat	Petugas Pendaftaran
RME_F_005	Melihat Jumlah Kunjungan Rawat Jalan	Petugas Pendaftaran
RME_F_006	Cetak Jumlah Kunjungan Rawat Jalan	Petugas Pendaftaran
RME_F_007	Melihat Peringkat Diagnosa	Petugas Pendaftaran
RME_F_008	Cetak Peringkat Diagnosa	Petugas Pendaftaran

RME_F_009	Mengubah Status Konfirmasi Pendaftaran	Petugas Pendaftaran
RME_F_010	Membuat Rekam Medis Baru	Petugas Rekam Medis
RME_F_011	Melihat Data Pribadi Pasien	Petugas Rekam Medis, Perawat
RME_F_012	Mengubah Data Pribadi Pasien	Petugas Rekam Medis
RME_F_013	Menghapus Rekam Medis	Petugas Rekam Medis
RME_F_014	Melihat <i>List</i> Pendaftaran Pasien Per Poli	Perawat
RME_F_015	Mengubah Status Antrian Pasien	Perawat
RME_F_016	Melihat Rekam Medis Pasien	Perawat
RME_F_017	Melihat Riwayat Pengkajian Klinis Medis	Perawat
RME_F_018	Melihat <i>Detail</i> Data Pengkajian Klinis Medis Rawat Jalan	Perawat
RME_F_019	Membuat Pengkajian Klinis Medis Rawat Jalan Baru	Perawat
RME_F_020	Membuat Data Rawat Jalan Umum Baru	Perawat
RME_F_021	Membuat Data Rawat Jalan Gigi Baru	Perawat
RME_F_022	Melihat <i>Detail</i> Data Rawat Jalan Umum	Perawat
RME_F_023	Melihat <i>Detail</i> Data Rawat Jalan Gigi	Perawat
RME_F_024	Menambah Data Rawat Jalan Umum	Perawat
RME_F_025	Menambah Data Rawat Jalan Gigi	Perawat
RME_F_026	Melihat Riwayat Kunjungan Umum	Perawat
RME_F_027	Melihat Riwayat Kunjungan Gigi	Perawat
RME_F_028	Melihat Riwayat Farmasi	Perawat
RME_F_029	Menambah Data Terapi Farmasi	Perawat
RME_F_030	Melihat Detail Terapi Farmasi	Perawat
RME_F_031	Melihat Riwayat Diagnostik	Perawat
RME_F_032	Melihat Hasil Pemeriksaan Diagnostik	Perawat
RME_F_033	Menambah Permintaan Pemeriksaan Diagnostik	Perawat
RME_F_034	Melihat Riwayat Pemberian Informasi Dan Edukasi Rawat Jalan	Perawat
RME_F_035	Melihat Detail Data Pemberian Informasi Dan Edukasi Rawat Jalan	Perawat
RME_F_036	Menambah Data Pemberian Informasi Dan Edukasi Rawat Jalan	Perawat
RME_F_037	Melihat Daftar Permintaan Pemeriksaan Diagnostik	Petugas Laboratorium

RME_F_038	Melihat Detail Permintaan Pemeriksaan Diagnostik	Petugas Laboratorium
RME_F_039	Menambah Hasil Pemeriksaan Diagnostik	Petugas Laboratorium

4.5.2 Kebutuhan Non-Fungsional

Kebutuhan fungsional sistem diberi kode dengan penamaan RME_NF_XXX. RME merupakan singkatan dari nama sistem yang dibuat yaitu Rekam Medis Elisa, NF merupakan tanda bahwa kebutuhan tersebut adalah kebutuhan non-fungsional, dan XXX merupakan nomor dari kebutuhan non-fungsional.

Kebutuhan non-fungsional pada sistem ini dijabarkan pada tabel 4.4 berikut.

Tabel 4.4 Daftar Kebutuhan Non-fungsional

No.	Kode	Parameter	Deskripsi
1.	RME_NF_001	<i>Compatibility</i>	Sistem dapat dijalankan di berbagai <i>browser</i> seperti Mozilla Firefox, Google Chrome, dan Microsoft Edge.

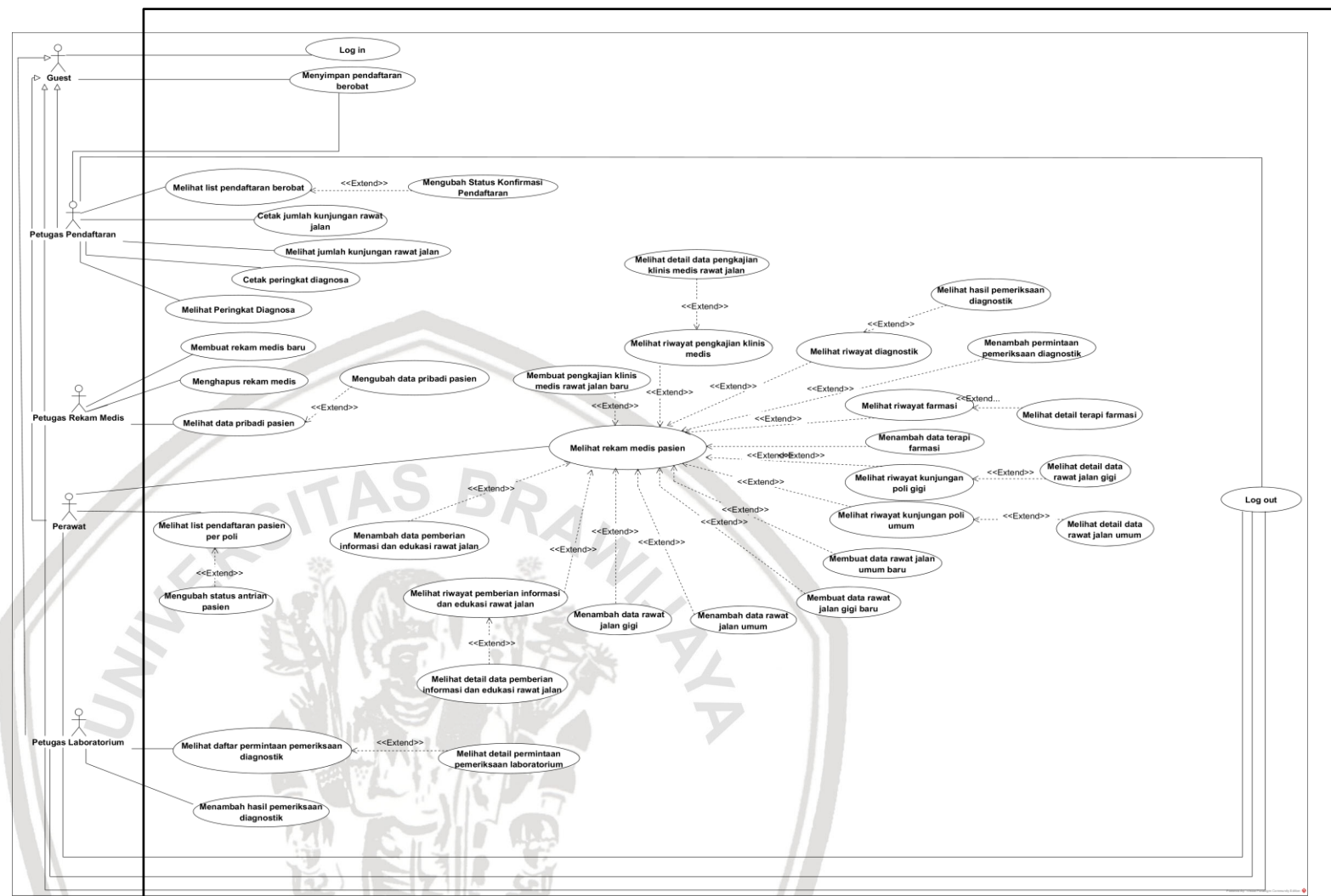
Pada sistem rekam medis elektronik dan pendaftaran pasien *online*, didapatkan kebutuhan non-fungsional *compatibility* untuk memastikan bahwa situs web yang dihasilkan dapat dijalankan pada berbagai jenis *browser* yang digunakan oleh aktor.

4.6 Pemodelan Use Case

Pemodelan *use case* terdiri dari dua tahap yaitu yang pertama membuat *use case diagram* yang menunjukkan hubungan aktor dengan perilaku sistem. Yang kedua adalah penjabaran dari masing-masing *use case* yaitu *Use Case Scenario*.

4.6.1 Use Case Diagram

Use case diagram untuk pembangunan sistem rekam medis elektronik dan pendaftaran pasien dengan studi kasus Klinik Medis Elisa dijabarkan pada Gambar 4.4.



Gambar 4.4 Use Case Diagram

Use Case Diagram pada gambar 4.4 menjabarkan hubungan antara aktor dengan *use case*. Terdapat lima aktor pada sistem ini yaitu *guest* yang dapat melakukan *log in* dan pendaftaran berobat, petugas pendaftaran yang berhubungan dengan fungsionalitas pendaftaran, petugas rekam medis yang mengakses fungsionalitas yang berhubungan dengan rekam medis. Kemudian terdapat aktor perawat yang mengakses fungsi yang berhubungan dengan data rawat jalan pasien, pengkajian klinis medis, dan pemberian edukasi kepada pasien. Terdapat pula aktor petugas laboratorium yang mengakses fungsi yang berhubungan dengan pemeriksaan diagnostik pasien.

4.6.2 Use Case Scenario

Use Case Scenario adalah penjabaran alur dari masing-masing aktivitas pada *Use Case Diagram* yang dituangkan ke dalam sebuah tabel untuk setiap *use case*. Jumlah dari tabel *use case scenario* disesuaikan dengan jumlah *use case*. Pada sistem ini terdapat 45 *use case scenario*.

Skenario pendaftaran berobat dengan aktor *guest* dan petugas pendaftaran dijabarkan pada Tabel 4.5 berikut.

Tabel 4.5 Use Case Scenario Daftar Berobat

Menyimpan Pendaftaran Berobat	
Kode Kebutuhan	RME_F_001
Objective	Melakukan penyimpanan pendaftaran berobat khusus untuk pasien yang sudah pernah berobat ke Klinik Elisa.
Aktor	<i>Guest</i> , petugas pendaftaran
Pre-Condition	<ol style="list-style-type: none"> 1. Untuk <i>guest</i>, telah membuka web pendaftaran 2. Untuk petugas pendaftaran, sudah melakukan <i>log in</i> dan berada di halaman utama petugas pendaftaran
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "Daftar Berobat" 2. Sistem menampilkan <i>form</i> pendaftaran berobat 3. Aktor mengisi <i>form</i> pendaftaran berobat secara lengkap 4. Aktor menekan tombol "Simpan" 5. Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> i. Jika terdapat kesalahan pada validasi, sistem kembali menampilkan <i>form</i> pendaftaran berobat dengan pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data.
Post-Condition	<ol style="list-style-type: none"> 1. Untuk <i>guest</i>, sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan menampilkan halaman baru yang berisi notifikasi bahwa pendaftaran berhasil dan nomor pendaftaran.

Menyimpan Pendaftaran Berobat	
Kode Kebutuhan	RME_F_001
	2. Untuk petugas pendaftaran, sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan menampilkan halaman utama petugas pendaftaran yang berisi tabel pendaftaran pasien yang telah diperbaharui.

Skenario *log in* dengan aktor *guest* dijabarkan pada Tabel 4.6 berikut.

Tabel 4.6 Use Case Scenario Log In

Log In	
Kode Kebutuhan	RME_F_002
Objective	Masuk kedalam sistem dan menggunakan fungsionalitas yang ada pada sistem berdasarkan jenis pengguna
Aktor	<i>Guest</i>
Pre-Condition	Aktor telah membuka web dan memilih menu “Login”
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form log in</i> 2. Aktor mengisi <i>form log in</i> secara lengkap 3. Aktor menekan tombol “log in” 4. Sistem melakukan validasi dan memeriksa ke <i>database</i> apakah <i>username</i> dan <i>password</i> terdaftar
Alternative Flow	<ol style="list-style-type: none"> 4. Sistem melakukan validasi dan memeriksa ke <i>database</i> apakah <i>username</i> dan <i>password</i> terdaftar <ol style="list-style-type: none"> i. Jika validasi gagal, maka sistem akan menampilkan kembali <i>form log in</i> dengan pesan kesalahan pada <i>field</i> yang belum diisi. ii. Jika <i>username</i> dan <i>password</i> salah, sistem akan menampilkan peringatan bahwa kombinasi <i>username/password</i> salah dan kembali pada halaman <i>log in</i>.
Post-Condition	Sistem menampilkan halaman utama berdasarkan tipe pengguna

Skenario Melihat *list* Pendaftaran Berobat dengan aktor petugas pendaftaran dijabarkan pada Tabel 4.7 berikut.

Tabel 4.7 Use Case Scenario Melihat List Pendaftaran Berobat

Melihat List Pendaftaran Berobat	
Kode Kebutuhan	RME_F_004
Objective	Melihat data berobat pasien yang telah terdaftar
Aktor	Petugas pendaftaran
Pre-Condition	Aktor telah <i>log in</i> ke sistem sebagai petugas pendaftaran
Main Flow	Sistem mengambil data pendaftaran berobat dari <i>database</i>

Melihat List Pendaftaran Berobat	
Alternative Flow	Jika pendaftaran berobat masih kosong, sistem menampilkan tulisan “No data available in table” pada badan tabel.
Post-Condition	Sistem menampilkan halaman yang berisi tabel data pasien yang telah mendaftar untuk berobat

Skenario Melihat Jumlah Kunjungan Rawat Jalan dengan aktor petugas pendaftaran dijabarkan pada Tabel 4.8 berikut.

Tabel 4.8 Use Case Scenario Melihat Jumlah Kunjungan Rawat Jalan

Melihat Jumlah Kunjungan Rawat Jalan	
Kode Kebutuhan	RME_F_005
Objective	Melihat jumlah kunjungan rawat jalan berdasarkan poli dan bulan kunjungan
Aktor	Petugas Pendaftaran
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas pendaftaran 2. Sistem menampilkan halaman <i>list</i> pendaftaran berobat pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Jumlah Kunjungan” pada halaman utama. 2. Aktor memilih poli, bulan, dan tahun. 3. Sistem mengambil data jumlah kunjungan rawat jalan dari <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> 3. Jika data jumlah kunjungan pada bulan, poli, dan tahun yang dipilih kosong, sistem menampilkan <i>alert</i> bahwa data tidak ditemukan.
Post-Condition	Sistem menampilkan halaman yang berisi tabel jumlah kunjungan pasien setiap poli pada bulan yang telah dipilih.

Skenario Melihat Peringkat Diagnosa dengan aktor petugas pendaftaran dijabarkan pada Tabel 4.9 berikut.

Tabel 4.9 Use Case Scenario Melihat Peringkat Diagnosa

Melihat Peringkat Diagnosa	
Kode Kebutuhan	RME_F_007
Objective	Melihat diagnosa terbanyak berdasarkan poli dan bulan kunjungan.
Aktor	Petugas Pendaftaran
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas pendaftaran 2. Sistem menampilkan halaman <i>list</i> pendaftaran berobat pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Peringkat Diagnosa” pada halaman utama dan memilih bulan kunjungan. 2. Aktor memilih poli, bulan, dan tahun. 3. Sistem mengambil data dari <i>database</i>

Melihat Peringkat Diagnosa	
Alternative Flow	3. Jika peringkat diagnosa pada poli, bulan, dan tahun yang dipilih kosong, maka sistem menampilkan tabel dengan tulisan “no data available”
Post-Condition	Sistem menampilkan halaman yang berisi peringkat diagnosa pada bulan , poli, dan tahun yang telah dipilih.

Skenario Cetak Jumlah Kunjungan Rawat Jalan dengan aktor petugas pendaftaran dijabarkan pada Tabel 4.10 berikut.

Tabel 4.10 Use Case Scenario Cetak Jumlah Kunjungan Rawat Jalan

Cetak Jumlah Kunjungan Rawat Jalan	
Kode Kebutuhan	RME_F_006
Objective	Mencetak data jumlah kunjungan rawat jalan berdasarkan poli dan bulan kunjungan
Aktor	Petugas Pendaftaran
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas pendaftaran 2. Sistem telah menampilkan jumlah kunjungan pasien pada bulan, tahun, dan poli yang telah dipilih
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Cetak” pada halaman yang menampilkan jumlah kunjungan rawat jalan 2. Sistem menampilkan diagram kunjungan rawat jalan pasien berdasarkan poli bulan, tahun yang dipilih beserta <i>layout</i> untuk mencetak 3. Aktor menekan tombol <i>print</i>
Alternative Flow	3. Jika aktor menekan tombol “Batal” maka akan kembali ke halaman yang menampilkan diagram jumlah kunjungan rawat jalan
Post-Condition	Sistem mencetak jumlah kunjungan pasien kemudian kembali ke halaman jumlah kunjungan pasien

Skenario Cetak Peringkat Diagnosa dengan aktor petugas pendaftaran dijabarkan pada Tabel 4.11 berikut.

Tabel 4.11 Use Case Scenario Cetak Peringkat Diagnosa

Cetak Peringkat Diagnosa	
Kode Kebutuhan	RME_F_008
Objective	Mencetak data peringkat diagnosa berdasarkan bulan dan poli
Aktor	Petugas Pendaftaran
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas pendaftaran 2. Sistem telah menampilkan peringkat diagnosa pada bulan, tahun, dan, poli yang dipilih

Cetak Peringkat Diagnosa	
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Cetak” pada halaman peringkat diagnosa 2. Sistem menampilkan halaman berisi diagram peringkat diagnosa berdasarkan poli, bulan, dan tahun yang telah dipilih beserta <i>layout</i> untuk mencetak 3. Aktor menekan tombol <i>print</i>
Alternative Flow	<ol style="list-style-type: none"> 3. Jika aktor menekan tombol “Batal” maka akan kembali ke halaman yang menampilkan diagram peringkat diagnosa
Post-Condition	Sistem mencetak halaman diagnosa terbanyak kemudian kembali ke halaman peringkat diagnosa

Skenario Mengubah Status Konfirmasi Pendaftaran dengan aktor petugas pendaftaran dijabarkan pada Tabel 4.12 berikut.

Tabel 4.12 Use Case Scenario Mengubah Status Konfirmasi Pendaftaran

Mengubah Status Antrian Pasien	
Kode Kebutuhan	RME_F_009
Objective	Mengubah status konfirmasi pendaftaran berobat pada tabel pendaftaran berobat
Aktor	Petugas pendaftaran
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas pendaftaran 2. Sistem telah menampilkan halaman yang berisi tabel pendaftaran berobat pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Konfirmasi” pada tabel pendaftaran berobat pasien 2. Sistem menyimpan perubahan pada <i>database</i>
Alternative Flow	-
Post-Condition	Status antrian pasien telah berubah dari yang awalnya “Belum Konfirmasi” menjadi “Sudah Konfirmasi”

Skenario Membuat Rekam Medis Baru dengan aktor petugas rekam medis dijabarkan pada Tabel 4.13 berikut.

Tabel 4.13 Case Scenario Membuat Rekam Medis Baru

Membuat Rekam Medis Baru	
Kode Kebutuhan	RME_F_010
Objective	Membuat rekam medis pasien rawat jalan baru
Aktor	Petugas rekam medis
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas rekam medis 2. Sistem menampilkan halaman utama petugas rekam medis

Membuat Rekam Medis Baru	
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu “Buat Rekam Medis Baru” 2. Sistem menampilkan <i>form</i> untuk membuat rekam medis baru 3. Aktor mengisi <i>form</i> rekam medis baru secara lengkap 4. Aktor menekan tombol “Simpan” 5. Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> i. Jika terdapat kesalahan pada validasi, sistem kembali menampilkan <i>form</i> rekam medis baru dengan pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data.
Post-Condition	Sistem menampilkan <i>alert</i> bahwa pembuatan rekam medis berhasil

Skenario Melihat data pribadi pasien dengan aktor petugas rekam medis dijabarkan pada Tabel 4.14 berikut.

Tabel 4.14 Use Case Scenario Melihat Data Pribadi Pasien

Melihat Data Pribadi Rekam Medis	
Kode Kebutuhan	RME_F_011
Objective	Melihat data pribadi pasien pada rekam medis
Aktor	Petugas rekam medis, perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas rekam medis atau perawat 2. Untuk petugas rekam medis, telah menampilkan halaman utama petugas rekam medis 3. Untuk perawat, telah memilih poli tujuan 4. Untuk perawat, telah menampilkan data pendaftaran berobat pasien 5. Untuk perawat, telah menekan tombol “Cari rekam medis” pada halaman pendaftaran berobat pasien 6. Untuk perawat, sistem telah menampilkan halaman pencarian rekam medis
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengetikkan nama pasien pada kolom pencarian 2. Sistem mencocokkan dengan data rekam medis pada <i>database</i> 3. Sistem menampilkan data rekam medis yang cocok dengan nama pasien 4. Aktor menekan tombol “Buka” pada halaman hasil pencarian rekam medis pasien 5. Sistem mengambil data dari <i>database</i>

Melihat Data Pribadi Rekam Medis	
Alternative Flow	3. Sistem menampilkan pesan bahwa data rekam medis tidak ditemukan
Post-Condition	Sistem menampilkan data pribadi pasien

Skenario Mengubah Data Pribadi Pasien dengan aktor petugas rekam medis dijabarkan pada Tabel 4.15 berikut.

Tabel 4.15 Use Case Scenario Mengubah Data Pribadi Pasien

Mengubah Data Pribadi Pasien	
Kode Kebutuhan	RME_F_012
Objective	Mengubah data pribadi pasien pada rekam medis
Aktor	Petugas rekam medis
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas rekam medis 2. Aktor telah membuka <i>file</i> rekam medis pasien. 3. Sistem telah menampilkan data pribadi pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol "Edit" 2. Sistem menampilkan <i>form</i> berisi data pribadi pasien 3. Aktor melakukan perubahan dengan mengganti isi <i>field</i> pada <i>form</i> data pribadi pasien 4. Aktor menekan tombol "Simpan" 5. Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> i. Jika terdapat kesalahan pada validasi, sistem kembali menampilkan <i>form</i> data pribadi pasien dengan pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data.
Post-Condition	Sistem menampilkan <i>alert</i> bahwa pengubahan data rekam medis berhasil dan kembali pada halaman data pribadi pasien

Skenario Menghapus Rekam Medis dengan aktor petugas rekam medis dijabarkan pada Tabel 4.16 berikut.

Tabel 4.16 Use Case Scenario Menghapus Rekam Medis

Menghapus Rekam Medis	
Kode Kebutuhan	RME_F_013
Objective	Menghapus rekam medis pasien
Aktor	Petugas rekam medis

Menghapus Rekam Medis	
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai petugas rekam medis 2. Aktor berada pada halaman utama petugas rekam medis
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengetikkan nama pasien pada kolom pencarian 2. Sistem menampilkan hasil pencarian rekam medis pasien 3. Aktor memilih tombol “Hapus” pada data yang dipilih 4. Aktor memilih menu “Oke” pada kotak dialog 5. Sistem menyimpan perubahan pada <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> 4. Jika aktor memilih menu “Batal” pada kotak dialog, maka akan kembali ke halaman pencarian rekam medis 5. Jika koneksi dengan basis data terputus, sistem menampilkan pesan kesalahan koneksi basis data
Post-Condition	Sistem menampilkan <i>alert</i> bahwa penghapusan rekam medis berhasil kemudian kembali ke halaman pencarian rekam medis

Skenario Melihat *List* Pendaftaran Pasien Per Poli dengan aktor perawat dijabarkan pada Tabel 4.17 berikut.

Tabel 4.17 Use Case Scenario Melihat *List* Pendaftaran Pasien Per Poli

Melihat <i>List</i> Pendaftaran Pasien Per Poli	
Kode Kebutuhan	RME_F_014
Objective	Melihat <i>list</i> pendaftaran berobat pasien berdasarkan poli tujuan
Aktor	Perawat
Pre-Condition	Aktor telah <i>log in</i> ke sistem sebagai perawat
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman yang berisi menu poli umum poli gigi, atau kesehatan ibu dan anak 2. Aktor memilih poli 3. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman yang berisi data pasien yang telah mendaftar untuk berobat pada poli yang telah dipilih

Skenario Mengubah Status Antrian Pasien dengan aktor perawat dijabarkan pada Tabel 4.18 berikut.

Tabel 4.18 Use Case Scenario Mengubah Status Antrian Pasien

Mengubah Status Antrian Pasien	
Kode Kebutuhan	RME_F_015
Objective	Mengubah status antrian pasien pada tabel pendaftaran berobat

Mengubah Status Antrian Pasien	
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah memilih poli tujuan 3. Sistem telah menampilkan halaman <i>list</i> pendaftaran pasien berdasarkan poli yang telah dipilih
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Ubah” pada tabel antrian pelayanan pasien 2. Sistem menyimpan perubahan pada <i>database</i>
Alternative Flow	-
Post-Condition	Status antrian pasien telah berubah dari yang awalnya “Belum Dilayani” menjadi “Sudah Dilayani”

Skenario Melihat Rekam Medis Pasien dengan aktor perawat dijabarkan pada Tabel 4.19 berikut.

Tabel 4.19 Use Case Scenario Melihat Rekam Medis Pasien

Melihat Rekam Medis Pasien	
Kode Kebutuhan	RME_F_016
Objective	Melihat isi <i>file</i> rekam medis pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah memilih poli tujuan dan menampilkan halaman pendaftaran berobat pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor telah menekan tombol “Cari rekam medis” pada halaman pendaftaran berobat pasien 2. Sistem menampilkan halaman pencarian rekam medis 3. Aktor mengetikkan nama pasien pada kolom pencarian 4. Sistem mencocokkan dengan data rekam medis pada <i>database</i> 5. Sistem menampilkan data rekam medis yang cocok dengan nama pasien 6. Aktor menekan tombol “Buka” pada halaman hasil pencarian rekam medis pasien 7. Sistem mengambil data dari <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> 4. Jika data rekam medis tidak ada pada <i>database</i>, maka sistem menampilkan pesan bahwa data rekam medis tidak ditemukan
Post-Condition	Sistem menampilkan halaman yang berisi rekam medis pasien yang terdiri dari data pribadi pasien, pengkajian klinis medis, kesehatan umum, kesehatan gigi dan mulut, edukasi rawat jalan, <i>tab</i> riwayat medis

Skenario Melihat Riwayat Pengkajian Klinis Medis dengan aktor perawat dijabarkan pada Tabel 4.20 berikut.

Tabel 4.20 Use Case Scenario Melihat Riwayat Pengkajian Klinis Medis

Melihat Riwayat Pengkajian Klinis Medis	
Kode Kebutuhan	RME_F_017
Objective	Melihat riwayat pengkajian klinis medis rawat jalan pasien yang telah tersimpan
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih <i>tab</i> Pengkajian Klinis Medis 2. Sistem mengambil data dari <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> 2. Untuk pasien baru, sistem menampilkan tabel riwayat pengkajian klinis medis yang masih kosong
Post-Condition	Sistem menampilkan halaman yang berisi tabel riwayat pengkajian klinis medis rawat jalan pasien

Skenario Melihat Detail Data Pengkajian Klinis Medis Rawat Jalan dengan aktor perawat dijabarkan pada Tabel 4.21 berikut.

Tabel 4.21 Melihat Detail Data Pengkajian Klinis Medis Rawat Jalan

Melihat Detail Data Pengkajian Klinis Medis Rawat Jalan	
Kode Kebutuhan	RME_F_018
Objective	Melihat detail dari pengkajian klinis medis rawat jalan pasien yang telah dipilih
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor berada pada <i>tab</i> Pengkajian Klinis Medis 4. Sistem menampilkan tabel riwayat pengkajian klinis medis pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih data pengkajian klinis medis yang ingin dibuka pada tabel daftar pengkajian klinis medis rawat jalan 2. Aktor menekan tombol “Buka” pada data yang dipilih 3. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman yang berisi detail pengkajian klinis medis rawat jalan pasien

Skenario Membuat pengkajian klinis medis rawat jalan baru dengan aktor perawat dijabarkan pada Tabel 4.22 berikut.

Tabel 4.22 Use Case Scenario Membuat Pengkajian Klinis Medis Rawat Jalan Baru

Membuat Pengkajian Klinis Medis Rawat Jalan Baru	
Kode Kebutuhan	RME_F_019
Objective	Memasukkan data pengkajian klinis medis rawat jalan pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor berada pada <i>tab</i> Pengkajian Klinis Medis 4. Sistem menampilkan tabel riwayat pengkajian klinis medis pasien
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu “Tambah Data” 2. Sistem menampilkan <i>form</i> pengkajian klinis medis rawat jalan 3. Aktor mengisi <i>form</i> pengkajian klinis medis rawat jalan secara lengkap 4. Aktor menekan tombol “Simpan” 5. Sistem menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Jika koneksi dengan basis data terputus, sistem menampilkan pesan kesalahan koneksi basis data
Post-Condition	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis

Skenario Membuat Data Rawat Jalan Umum Baru dengan aktor perawat dijabarkan pada Tabel 4.23 berikut.

Tabel 4.23 Use Case Scenario Membuat Data Rawat Jalan Umum Baru

Membuat Data Rawat Jalan Umum Baru	
Kode Kebutuhan	RME_F_020
Objective	Memasukkan data rawat jalan umum untuk pasien baru pada rekam medis
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Kesehatan Umum
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol “Buat Data Baru” 2. Sistem menampilkan halaman <i>form</i> data rawat jalan umum 3. Aktor mengisi <i>form</i> rawat jalan umum secara lengkap

Membuat Data Rawat Jalan Umum Baru	
	<ol style="list-style-type: none"> Aktor menekan tombol "Simpan" Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> Jika terdapat kesalahan pada validasi, sistem menampilkan pesan kesalahan pada <i>field</i> yang salah. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data
Post-Condition	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis

Skenario Membuat Data Rawat Jalan Gigi Baru dengan aktor perawat dijabarkan pada Tabel 4.24 berikut.

Tabel 4.24 Use Case Scenario Membuat Data Rawat Jalan Gigi Baru

Membuat Data Rawat Jalan Gigi Baru	
Kode Kebutuhan	RME_F_021
Objective	Memasukkan data rawat jalan gigi untuk pasien baru pada rekam medis
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> Aktor telah <i>log in</i> ke sistem sebagai perawat Aktor telah melihat rekam medis pasien Aktor telah berada pada <i>tab</i> Kesehatan Gigi dan Mulut
Main Flow	<ol style="list-style-type: none"> Aktor menekan tombol "Buat Data Baru" Sistem menampilkan halaman <i>formulir</i> data rawat jalan gigi Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap Aktor mengunggah <i>file</i> foto odontogram Aktor menekan tombol "Simpan" Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> Jika terdapat kesalahan pada validasi, sistem menampilkan pesan kesalahan pada <i>field</i> yang salah. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data. Jika format <i>file</i> foto yang diunggah salah atau berukuran terlalu besar, sistem menampilkan pesan kesalahan.
Post-Condition	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis

Skenario Melihat Detail Data Rawat Jalan Umum dengan aktor perawat dijabarkan pada Tabel 4.25 berikut.

Tabel 4.25 Use Case Scenario Melihat Detail Data Rawat Jalan Umum

Melihat Detail Data Rawat Jalan Umum	
Kode Kebutuhan	RME_F_022
Objective	Melihat detail data rawat jalan umum yang dipilih
Aktor	Perawat
Pre-Condition	1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Kesehatan Umum
Main Flow	1. Sistem menampilkan tabel riwayat kunjungan umum 2. Aktor memilih menu “Buka” pada data kunjungan yang dipilih 3. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan data rawat jalan pasien yang telah dipilih

Skenario Melihat Detail Data Rawat Jalan Gigi dengan aktor perawat dijabarkan pada Tabel 4.26 berikut.

Tabel 4.26 Use Case Scenario Melihat Detail Data Rawat Jalan Gigi

Melihat Detail Data Rawat Jalan Gigi	
Kode Kebutuhan	RME_F_023
Objective	Melihat detail data rawat jalan gigi yang dipilih
Aktor	Perawat
Pre-Condition	1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Kesehatan Gigi dan Mulut
Main Flow	1. Sistem menampilkan tabel riwayat kunjungan gigi 2. Aktor memilih menu “Buka” pada data kunjungan yang dipilih 3. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan data rawat jalan pasien yang telah dipilih

Skenario Menambah Data Rawat Jalan Umum dengan aktor perawat dijabarkan pada Tabel 4.27 berikut.

Tabel 4.27 Use Case Scenario Menambah Data Rawat Jalan Umum

Menambah Data Rawat Jalan Umum	
Kode Kebutuhan	RME_F_024
Objective	Memasukkan data rawat jalan umum milik pasien lama pada rekam medis
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Kesehatan Umum
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan data rawat jalan umum terakhir 2. Aktor menekan tombol “Tambah” 3. Sistem menampilkan halaman <i>form</i> data rawat jalan umum 4. Aktor mengisi <i>form</i> data rawat jalan umum secara lengkap 5. Aktor menekan tombol “Simpan” 6. Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 6. Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> i. Jika terdapat kesalahan pada validasi, sistem menampilkan pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data
Post-Condition	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis

Skenario Menambah Data Rawat Jalan Gigi dengan aktor perawat dijabarkan pada Tabel 4.28 berikut.

Tabel 4.28 Use Case Scenario Menambah Data Rawat Jalan Gigi

Menambah Data Rawat Jalan Gigi	
Kode Kebutuhan	RME_F_025
Objective	Memasukkan data rawat jalan gigi milik pasien lama pada rekam medis
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Kesehatan Gigi
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan data rawat jalan gigi terakhir 2. Aktor menekan tombol “Tambah” 3. Sistem menampilkan halaman <i>formulir</i> data rawat jalan gigi 4. Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap

Menambah Data Rawat Jalan Gigi	
	5. Aktor mengunggah <i>file</i> foto odontogram 6. Aktor menekan tombol “Simpan” 7. Sistem melakukan validasi dan menyimpan data
Alternative Flow	7. Kegagalan validasi dan penyimpanan : i. Jika terdapat kesalahan pada validasi, sistem menampilkan pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data
Post-Condition	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis

Skenario Melihat Riwayat Kunjungan Umum dengan aktor perawat dijabarkan pada Tabel 4.29 berikut.

Tabel 4.29 Use Case Scenario Melihat Riwayat Kunjungan Umum

Melihat Riwayat Kunjungan Umum	
Kode Kebutuhan	RME_F_026
Objective	Melihat data riwayat kunjungan pasien pada pelayanan umum selama berobat di Klinik Elisa
Aktor	Perawat
Pre-Condition	1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Kesehatan Umum
Main Flow	Sistem mengambil data dari <i>database</i>
Alternative Flow	Untuk pasien baru, tabel riwayat kunjungan masih kosong
Post-Condition	Sistem menampilkan halaman yang berisi data riwayat kunjungan pasien dalam bentuk tabel

Skenario Melihat Riwayat Kunjungan Gigi dengan aktor perawat dijabarkan pada Tabel 4.30 berikut.

Tabel 4.30 Use Case Scenario Melihat Riwayat Kunjungan Gigi

Melihat Riwayat Kunjungan Gigi	
Kode Kebutuhan	RME_F_027
Objective	Melihat data riwayat kunjungan pasien pada pelayanan gigi selama berobat di Klinik Elisa
Aktor	Perawat
Pre-Condition	1. Aktor telah <i>log in</i> ke sistem sebagai perawat

Melihat Riwayat Kunjungan Gigi	
	<ol style="list-style-type: none"> Aktor telah melihat rekam medis pasien Aktor telah berada pada <i>tab</i> Kesehatan Gigi dan Mulut
Main Flow	Sistem mengambil data dari <i>database</i>
Alternative Flow	Untuk pasien baru, tabel riwayat kunjungan masih kosong
Post-Condition	Sistem menampilkan halaman yang berisi data riwayat kunjungan gigi dalam bentuk tabel

Skenario Melihat Riwayat Farmasi dengan aktor perawat dijabarkan pada Tabel 4.31 berikut.

Tabel 4.31 Use Case Scenario Melihat Riwayat Farmasi

Melihat Riwayat Farmasi	
Kode Kebutuhan	RME_F_028
Objective	Melihat data riwayat terapi farmasi pasien yang telah tersimpan
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> Aktor telah <i>log in</i> ke sistem sebagai perawat Aktor telah melihat rekam medis pasien Aktor telah berada pada <i>tab</i> Riwayat Medis
Main Flow	<ol style="list-style-type: none"> Aktor memilih menu “Riwayat Obat” pada <i>tab</i> Riwayat Medis Sistem mengambil data dari <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> Untuk pasien baru, sistem menampilkan tabel riwayat farmasi yang masih kosong
Post-Condition	Sistem menampilkan halaman yang berisi data riwayat terapi farmasi pasien dalam bentuk tabel

Skenario Menambah Data Terapi Farmasi dengan aktor perawat dijabarkan pada Tabel 4.32 berikut.

Tabel 4.32 Use Case Scenario Menambah Data Terapi Farmasi

Menambah Data Terapi Farmasi	
Kode Kebutuhan	RME_F_029
Objective	Menambah data terapi farmasi pasien pada rekam medis.
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> Aktor telah <i>log in</i> ke sistem sebagai perawat Aktor telah melihat rekam medis pasien Aktor telah berada pada <i>tab</i> Riwayat Medis Sistem menampilkan tabel riwayat obat

Menambah Data Terapi Farmasi	
Main Flow	<ol style="list-style-type: none"> 1. Pada tab Riwayat Obat, aktor menekan tombol “Tambah Data” 2. Sistem menampilkan <i>form</i> untuk menambah data farmasi 3. Aktor mengisi <i>form</i> terapi farmasi secara lengkap 4. Aktor menekan tombol “Simpan” 5. Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> i. Jika terdapat kesalahan pada validasi, sistem menampilkan kembali <i>form</i> terapi farmasi beserta pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data
Post-Condition	Sistem akan menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis.

Skenario Melihat Detail Terapi Farmasi dengan aktor perawat dijabarkan pada Tabel 4.33 berikut.

Tabel 4.33 Use Case Scenario Melihat Detail Terapi Farmasi

Melihat Detail Terapi Farmasi	
Kode Kebutuhan	RME_F_030
Objective	Melihat detail terapi farmasi yang telah disimpan
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Riwayat Medis
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan tabel riwayat obat 2. Aktor memilih data yang akan dilihat detailnya 3. Aktor menekan tombol “Detail” 4. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman yang berisi detail terapi farmasi yang telah dipilih

Skenario Melihat Riwayat Diagnostik dengan aktor perawat dijabarkan pada Tabel 4.34 berikut.

Tabel 4.34 *Use Case Scenario* Melihat Riwayat Diagnostik

Melihat Riwayat Diagnostik	
Kode Kebutuhan	RME_F_031
Objective	Melihat data riwayat terapi diagnostik yang telah tersimpan
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah membuka <i>file</i> rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Riwayat Medis
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu “ Riwayat Diagnostik” pada <i>tab</i> Riwayat Medis 2. Sistem mengambil data dari <i>database</i>
Alternative Flow	<ol style="list-style-type: none"> 1. Untuk pasien baru, sistem menampilkan tabel riwayat diagnostik yang masih kosong
Post-Condition	Sistem menampilkan halaman yang berisi data riwayat terapi diagnostik pasien dalam bentuk tabel

Skenario Menambah Permintaan Pemeriksaan Diagnostik dengan aktor perawat dijabarkan pada Tabel 4.35 berikut.

Tabel 4.35 *Use Case Scenario* Menambah Permintaan Pemeriksaan Diagnostik

Menambah Permintaan Pemeriksaan Diagnostik	
Kode Kebutuhan	RME_F_033
Objective	Melakukan penambahan data permintaan pemeriksaan diagnostik pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Riwayat Medis 4. Sistem menampilkan tabel riwayat diagnostik
Main Flow	<ol style="list-style-type: none"> 1. Pada <i>tab</i> riwayat diagnostik, aktor menekan tombol “Tambah Data” 2. Sistem menampilkan <i>form</i> untuk menambah data permintaan pemeriksaan diagnostik 3. Aktor mengisi <i>form</i> penambahan data permintaan pemeriksaan diagnostik secara lengkap 4. Aktor menekan tombol “Simpan” 5. Sistem menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Jika koneksi dengan basis data terputus, sistem menampilkan pesan kesalahan koneksi basis data

Menambah Permintaan Pemeriksaan Diagnostik	
Post-Condition	Sistem akan menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman yang menampilkan tabel riwayat diagnostik

Skenario Melihat Hasil Pemeriksaan Diagnostik dengan aktor perawat dijabarkan pada Tabel 4.36 berikut.

Tabel 4.36 Use Case Scenario Melihat Hasil Pemeriksaan Diagnostik Pasien

Melihat Hasil Pemeriksaan Diagnostik	
Kode Kebutuhan	RME_F_032
Objective	Melihat hasil pemeriksaan diagnostik pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Riwayat Medis 4. Sistem menampilkan tabel riwayat diagnostik
Main Flow	<ol style="list-style-type: none"> 1. Pada tabel riwayat diagnostik, aktor menekan tombol "Lihat Hasil" 2. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan halaman yang berisi hasil pemeriksaan diagnostik yang telah dipilih pada tabel

Skenario Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan dengan aktor perawat dijabarkan pada Tabel 4.37 berikut.

Tabel 4.37 Use Case Scenario Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan

Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan	
Kode Kebutuhan	RME_F_036
Objective	Menambahkan data pemberian informasi dan edukasi rawat jalan pada rekam medis pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Edukasi Rawat Jalan pada rekam medis pasien 4. Sistem menampilkan tabel riwayat pemberian informasi dan edukasi rawat jalan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "Tambah Data"

Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan	
	<ol style="list-style-type: none"> 2. Sistem menampilkan <i>form</i> Pemberian Informasi dan Edukasi Rawat Jalan 3. Aktor mengisi <i>form</i> secara lengkap 4. Aktor menekan tombol “Simpan” 5. Sistem melakukan validasi dan menyimpan data
Alternative Flow	<ol style="list-style-type: none"> 5. Kegagalan validasi dan penyimpanan : <ol style="list-style-type: none"> i. Jika terdapat kesalahan pada validasi, sistem menampilkan kembali <i>form</i> edukasi rawat jalan beserta pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data
Post-Condition	Sistem menampilkan <i>alert</i> bahwa pemberian informasi dan edukasi rawat jalan baru berhasil disimpan dan kembali pada halaman rekam medis pasien

Skenario Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan dengan aktor perawat dijabarkan pada Tabel 4.38 berikut.

Tabel 4.38 Use Case Scenario Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan

Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan	
Kode Kebutuhan	RME_F_034
Objective	Melihat data pemberian informasi dan edukasi rawat jalan pada rekam medis pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah membuka <i>file</i> rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Edukasi Rawat Jalan pada rekam medis pasien
Main Flow	Sistem mengambil data pemberian informasi dan edukasi rawat jalan pasien dari <i>database</i>
Alternative Flow	Untuk pasien baru, sistem menampilkan tabel riwayat pemberian edukasi rawat jalan yang masih kosong
Post-Condition	Sistem menampilkan tabel riwayat pemberian informasi dan edukasi rawat jalan

Skenario Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan dengan aktor perawat dijabarkan pada Tabel 4.39 berikut.

Tabel 4.39 Use Case Scenario Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan

Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan	
Kode Kebutuhan	RME_F_035
Deskripsi	Proses untuk melihat detail data pemberian informasi dan edukasi rawat jalan pada rekam medis pasien
Aktor	Perawat
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>log in</i> ke sistem sebagai perawat 2. Aktor telah melihat rekam medis pasien 3. Aktor telah berada pada <i>tab</i> Edukasi Rawat Jalan pada rekam medis pasien 4. Sistem menampilkan tabel riwayat pemberian informasi dan edukasi rawat jalan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu “Buka” pada tabel riwayat pemberian informasi dan edukasi rawat jalan 2. Sistem mengambil data dari <i>database</i>
Alternative Flow	-
Post-Condition	Sistem menampilkan detail dari data pemberian informasi dan edukasi rawat jalan yang dipilih

Skenario Melihat Daftar Permintaan Pemeriksaan Diagnostik dengan aktor petugas laboratorium dijabarkan pada Tabel 4.40 berikut.

Tabel 4.40 Use Case Scenario Melihat Daftar Permintaan Pemeriksaan Diagnostik

Melihat Daftar Permintaan Pemeriksaan Diagnostik	
Kode Kebutuhan	RME_F_037
Objective	Melihat daftar permintaan pemeriksaan diagnostik
Aktor	Petugas laboratorium
Pre-Condition	Aktor telah <i>log in</i> ke sistem sebagai petugas laboratorium
Main Flow	Sistem mengambil data dari <i>database</i>
Alternative Flow	Data pendaftaran berobat masih kosong, sistem menampilkan tulisan “No data available in table” pada badan tabel.
Post-Condition	Sistem menampilkan halaman yang berisi tabel daftar permintaan diagnostik

Skenario Melihat Detail Permintaan Pemeriksaan Diagnostik dengan aktor petugas laboratorium dijabarkan pada Tabel 4.41 berikut.

Tabel 4.41 Use Case Scenario Melihat Detail Permintaan Pemeriksaan Diagnostik

Melihat Detail Permintaan Pemeriksaan Diagnostik	
Kode Kebutuhan	RME_F_038
Objective	Melihat detail permintaan pemeriksaan diagnostik yang telah disimpan
Aktor	Petugas laboratorium
Pre-Condition	1. Aktor telah <i>log in</i> ke sistem sebagai petugas laboratorium 2. Sistem menampilkan daftar permintaan pemeriksaan diagnostik
Main Flow	1. Aktor memilih data yang akan dilihat detailnya 2. Aktor memilih menu “Buka” 3. Sistem mengambil data dari <i>database</i>
Alternative Flow	Data pendaftaran berobat masih kosong, sistem menampilkan tulisan “No data available in table” pada badan tabel.
Post-Condition	Sistem menampilkan halaman yang berisi detail permintaan pemeriksaan diagnostik yang telah dipilih

Skenario Menambah Hasil Pemeriksaan Diagnostik dengan aktor petugas laboratorium dijabarkan pada Tabel 4.42 berikut.

Tabel 4.42 Use Case Scenario Menambah Hasil Pemeriksaan Diagnostik

Menambah Hasil Pemeriksaan Diagnostik	
Kode Kebutuhan	RME_F_039
Objective	Menambahkan hasil pemeriksaan diagnostik ke rekam medis pasien
Aktor	Petugas laboratorium
Pre-Condition	1. Aktor telah <i>log in</i> ke sistem sebagai petugas laboatorium 2. Sistem menampilkan daftar permintaan pemeriksaan diagnostik
Main Flow	1. Aktor memilih menu “Input Hasil” pada tabel permintaan pemeriksaan diagnostik 2. Sistem menampilkan <i>form</i> hasil pemeriksaan diagnostik 3. Aktor mengisi <i>form</i> hasil pemeriksaan diagnostik secara lengkap 4. Aktor mengunggah <i>file</i> foto hasil pemeriksaan diagnostik 5. Aktor menekan tombol “Simpan” 6. Sistem melakukan validasi dan menyimpan data
Alternative Flow	6. Kegagalan validasi dan penyimpanan : i. Jika terdapat kesalahan pada validasi, sistem menampilkan kembali <i>form</i> hasil pemeriksaan diagnostik beserta pesan kesalahan pada <i>field</i> yang salah. ii. Jika koneksi basis data terputus, sistem menampilkan pesan kesalahan basis data

Menambah Hasil Pemeriksaan Diagnostik	
	iii. Jika format <i>file</i> foto yang diunggah salah atau ukuran <i>file</i> terlalu besar, maka sistem menampilkan pesan kesalahan
Post-Condition	Sistem menampilkan <i>alert</i> bahwa hasil pemeriksaan diagnostik berhasil disimpan dan kembali pada halaman daftar permintaan pemeriksaan diagnostik

Skenario *Log out* dengan aktor petugas pendaftaran, petugas rekam medis, perawat, dan petugas laboratorium dijabarkan pada Tabel 4.43 berikut.

Tabel 4.43 Use Case Scenario Log out

Logout	
Kode Kebutuhan	RME_F_003
Objective	Mengakhiri <i>session</i> yang sedang berlangsung untuk keluar menjadi <i>guest</i>
Aktor	Petugas pendaftaran, petugas rekam medis, perawat, petugas farmasi, petugas laboratorium
Pre-Condition	Aktor telah <i>log in</i> ke sistem sesuai jenis aktor
Main Flow	1. Aktor memilih tombol " <i>Log out</i> " pada halaman web 2. Sistem mengakhiri sesi
Alternative Flow	-
Post-Condition	Sistem kembali ke halaman utama web

BAB 5 PERANCANGAN DAN IMPLEMENTASI

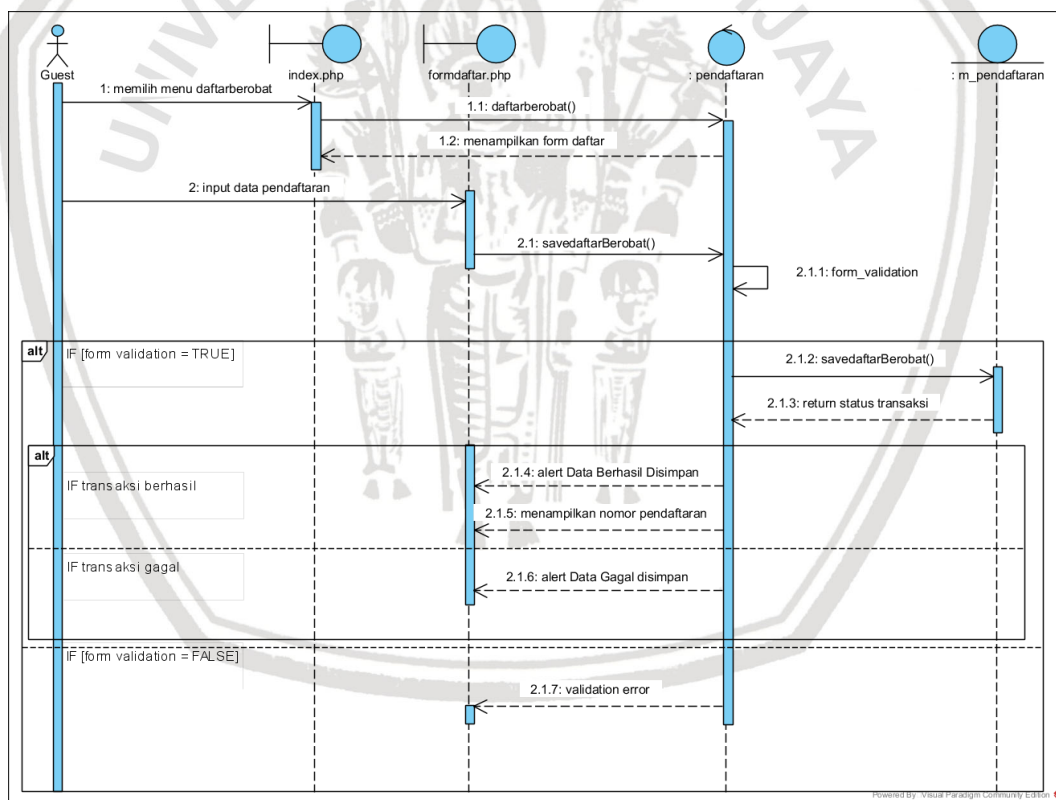
5.1 Perancangan

Perancangan dilakukan berdasarkan hasil rekayasa kebutuhan yang telah dilakukan sebelumnya. Perancangan digunakan sebagai acuan untuk implementasi pada proses pembangunan sistem. Perancangan dilakukan dalam beberapa tahap yaitu pembuatan *sequence diagram*, pembuatan *class diagram*, perancangan algoritme, dan perancangan antarmuka.

5.1.1 Sequence Diagram

Sequence diagram merepresentasikan pertukaran pesan antarobjek dalam garis waktu. Terdapat tiga sampel *sequence diagram* dalam penelitian ini yaitu *Sequence Diagram Menyimpan Pendaftaran Berobat*, *Sequence Diagram Membuat Rekam Medis Baru*, *Sequence Diagram Menambah Data Rawat Jalan Gigi*, dan *Sequence Diagram Menambah Hasil Pemeriksaan Diagnostik*.

Sequence Diagram Menyimpan Pendaftaran Berobat ditunjukkan pada gambar 5.1 berikut.

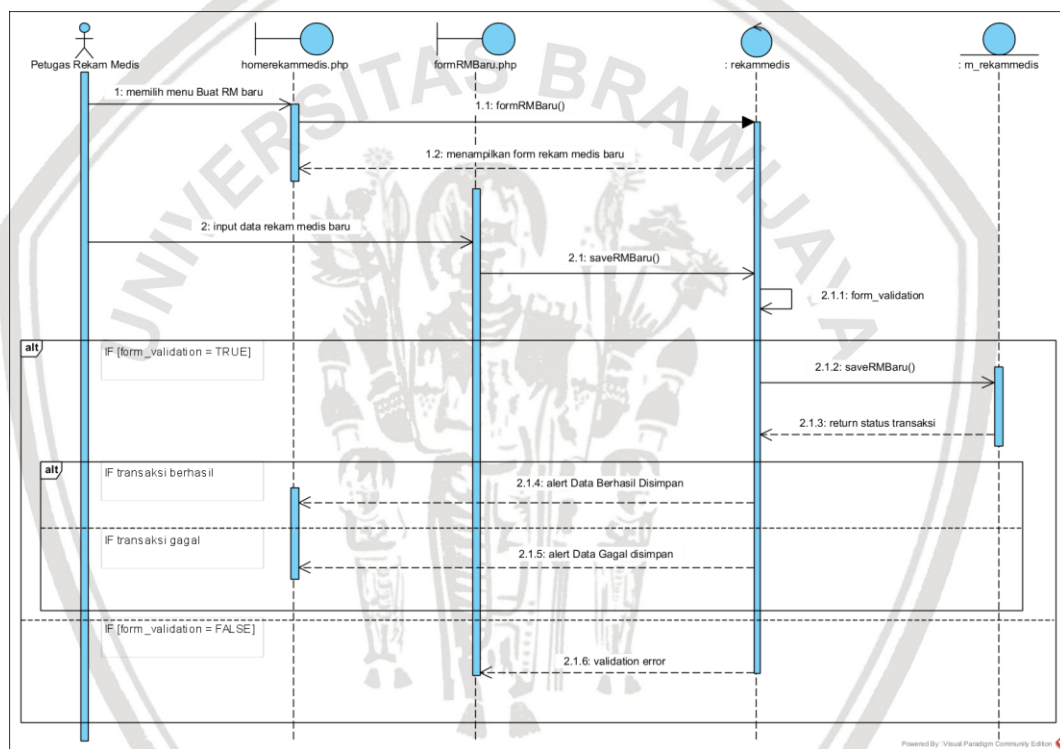


Gambar 5.1 Sequence Diagram Menyimpan Pendaftaran Berobat

Pada *sequence diagram* Menyimpan Pendaftaran Berobat, aktor yang terlibat adalah *guest*. Awalnya *guest* berada pada halaman *index* kemudian aktor memilih menu “Daftar Berobat”. Sistem akan menjalankan fungsi *daftarberobat()* pada *controller pendaftaran* yang kemudian akan menampilkan *form* pendaftaran.

Aktor melakukan *input data* pendaftaran pada *form* kemudian menekan tombol “Simpan”. Lalu sistem menjalankan fungsi `savedaftarBerobat()`. Lalu sistem melakukan validasi masukan pada *form*, jika masukan aktor sudah benar maka sistem akan yang akan berlanjut ke fungsi `savedaftarBerobat()` pada class `m_pendaftaran` untuk menyimpan data ke *database* dan sistem akan menampilkan nomor pendaftaran. Jika transaksi berhasil akan menampilkan pesan bahwa pendaftaran berhasil disimpan, jika transaksi gagal akan menampilkan pesan pendaftaran gagal disimpan. Jika masukan aktor pada *form* daftar belum benar maka sistem tidak dapat menyimpan data pada *database* dan menampilkan *error message*.

Sequence Diagram Membuat Rekam Medis Baru ditunjukkan pada gambar 5.2 berikut.

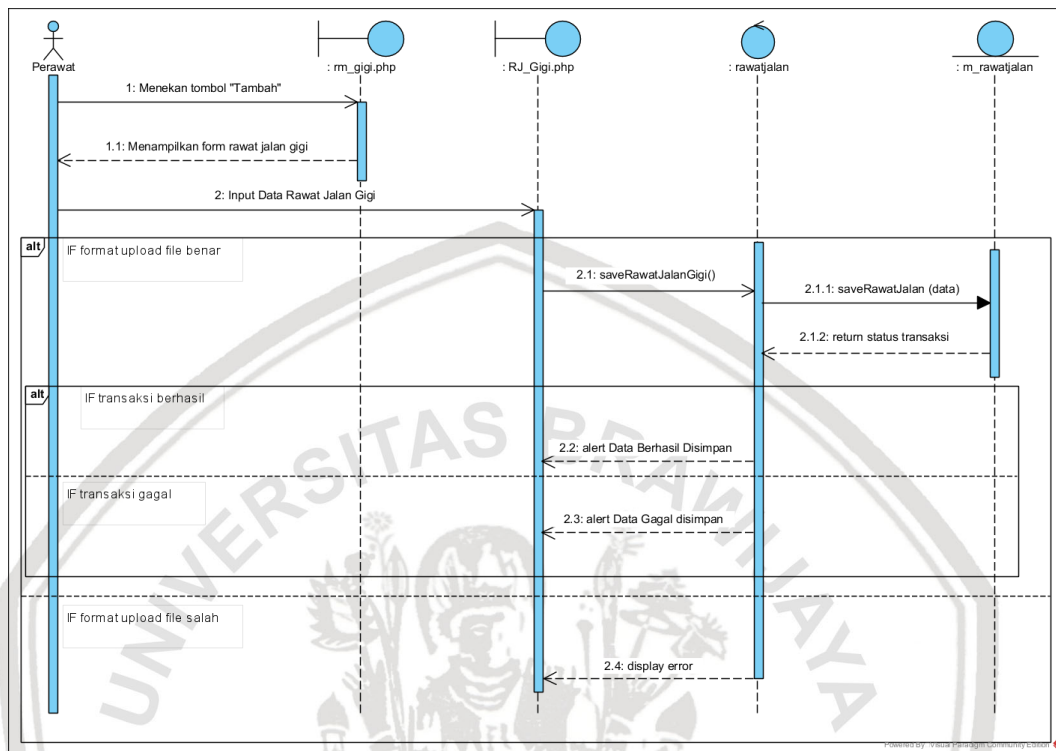


Gambar 5.2 Sequence Diagram Membuat Rekam Medis Baru

Pada *sequence diagram* Membuat Rekam Medis Baru, aktor yang terlibat adalah petugas rekam medis. Awalnya aktor berada pada halaman `homerekammedis`, lalu aktor memilih menu “Buat RM Baru” yang akan memanggil fungsi `formRMBaru()` pada *controller* `rekammedis` dan akan menampilkan *form* rekam medis baru. Lalu aktor memasukkan data rekam medis baru. Setelah selesai input data, kemudian sistem menjalankan validasi *form* untuk mengetahui apakah ada kesalahan pada saat memasukkan data ke *form*. Jika tidak terdapat kesalahan, maka sistem akan menjalankan fungsi `saveRMBaru()` yang mengembalikan status transaksi. Jika transaksi berhasil akan menampilkan pesan bahwa rekam medis berhasil disimpan, jika transaksi gagal akan menampilkan pesan rekam medis

gagal disimpan. Jika terdapat *error* pada *form*, maka akan menampilkan pesan bahwa validasi *error* pada halaman *form* rekam medis baru.

Sequence Diagram Menambah Data Rawat Jalan Gigi ditunjukkan pada gambar 5.3 berikut.

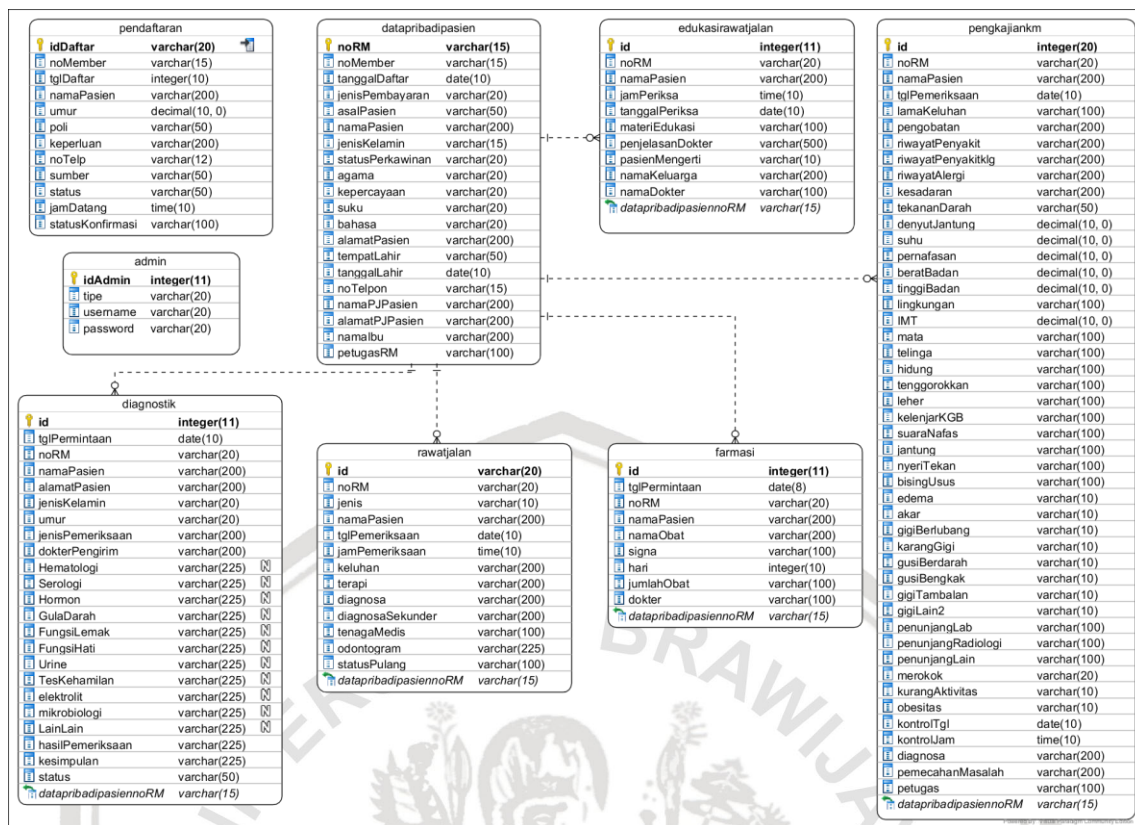


Gambar 5.3 Sequence Diagram Menambah Data Rawat Jalan Gigi

Pada *sequence diagram* Menambah Data Rawat Jalan Gigi aktor yang terlibat adalah perawat. Awalnya aktor berada pada halaman rekam medis gigi, lalu aktor menekan tombol “Tambah” dan akan menampilkan *form* rawat jalan gigi. Lalu aktor memasukkan data rawat jalan gigi yang didalamnya termasuk mengunggah file foto odontogram. Setelah selesai input data, kemudian sistem menjalankan fungsi `saveRawatJalanGigi()`. Kemudian akan memanggil fungsi `saveRawatJalan(data)` pada model. Jika transaksi berhasil akan menampilkan pesan bahwa data berhasil disimpan, jika transaksi gagal akan menampilkan data gagal disimpan. Jika aktor menunggah file foto dengan format atau ukuran yang tidak tepat maka akan menampilkan pesan *error*.

5.1.2 Perancangan Basis Data

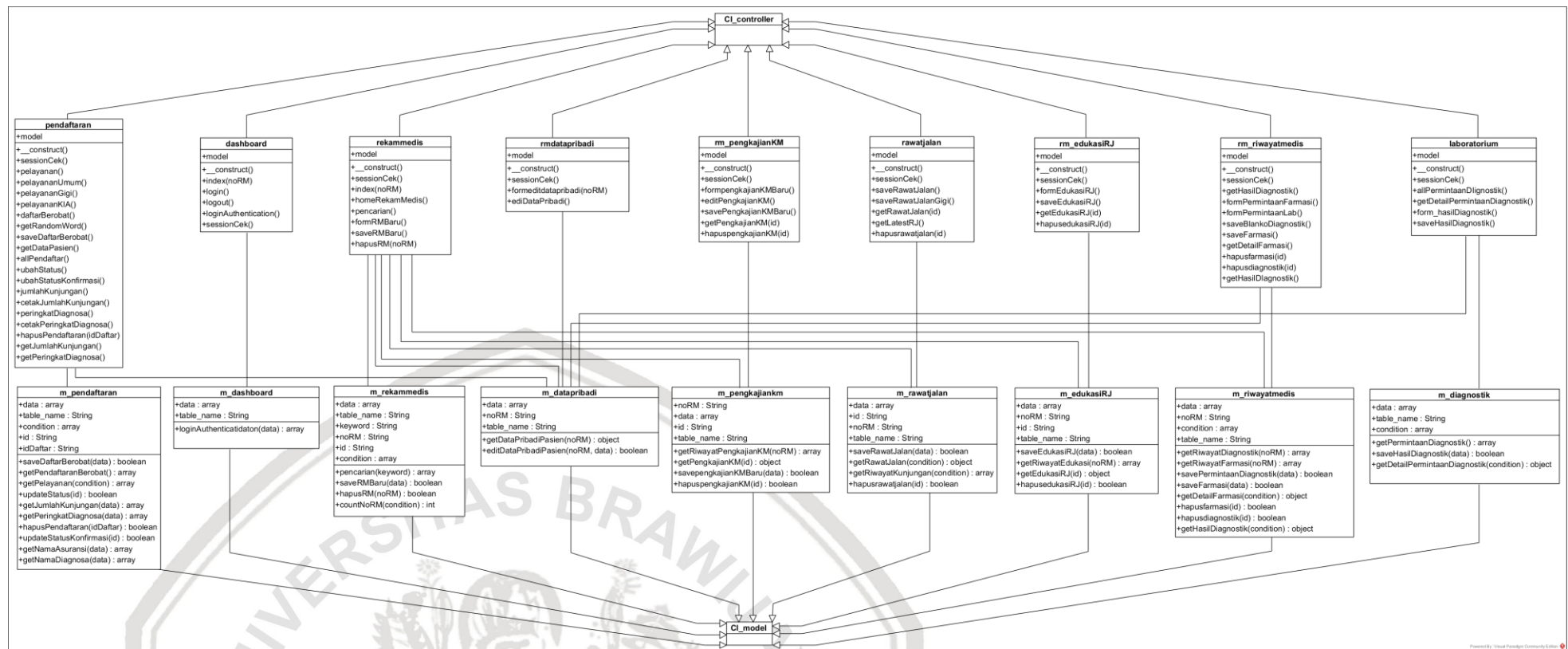
Perancangan basis data dilakukan menggunakan *physical data model* yang merupakan gambaran fisik dari tabel-tabel dan relasinya dalam *database* sesuai dengan analisis data yang telah dilakukan. *Physical Data Model* digambarkan pada gambar 5.4 berikut.



Gambar 5.4 Physical Data Model

5.1.3 Class Diagram

Class diagram adalah diagram yang merepresentasikan interaksi kelas-kelas dalam sistem yang dijabarkan pada gambar 5.5 berikut.



Gambar 5.5 Class Diagram

Pada *class diagram* terdapat 9 kelas yang merupakan turunan dari kelas *CI_contoller* yaitu kelas *dashboard* yang berisi fungsi-fungsi yang mengatur tentang halaman awal dan pendaftaran berobat untuk *guest*. Kemudian terdapat kelas pendaftaran yang berisi fungsi yang berkaitan dengan pendaftaran berobat pasien, kelas rekam medis, *rmdatapribadi*, *rm_pengkajianKM*, *rawatjalan*, *rm_riwayatmedis*, dan *rm_edukasirawatjalan* yang berisi fungsi-fungsi yang mengatur tentang rekam medis pasien. Kemudian terdapat kelas laboratorium yang memiliki fungsi yang berkaitan dengan pemeriksaan diagnostik.

Terdapat 9 kelas yang diturunkan dari kelas *CI_model* yaitu *m_dashboard* yang berisi fungsi yang berkaitan dengan proses *log in* dan *log out*, *m_pendaftaran* yang berisi fungsi yang mengakses ke *database* untuk pendaftaran berobat, kelas *m_rekammedis* yang berisi fungsi-fungsi yang berkaitan dengan rekam medis untuk akses ke *database*, *m_rawatjalan*, *m_educasiRJ*, *m_pengkajiankm*, *m_riwayatmedis*. Lalu terdapat kelas *m_diagnostik* yang berisi fungsi-fungsi yang berkaitan dengan pemeriksaan diagnostik.

5.1.4 Perancangan Algoritme

Tabel 5.1 berikut menunjukkan perancangan algoritme Menyimpan Pendaftaran Berobat menggunakan *pseudocode*.

Tabel 5.1 Pseudocode Menyimpan Pendaftaran Berobat

No	Pseudocode
1	Mulai
2	Input data pendaftaran
3	Validasi <i>Form</i>
4	IF <i>form validation</i> = FALSE
5	menampilkan <i>form</i> pendaftaran
6	menampilkan pesan kesalahan
7	ELSE
8	Mengambil nilai masukan user
9	Inisialisasi nomor pendaftaran
10	Inisialisasi tanggal pendaftaran
11	IF login = TRUE
12	Inisialisasi sumber = "datang langsung"
13	ELSE
14	Inisialisasi sumber = "aplikasi"
15	END IF
16	Insert data ke database
17	IF insert = TRUE
18	Menampilkan pesan "Data berhasil disimpan"
19	ELSE
20	Menampilkan pesan "Data gagal disimpan"
21	Redirect halaman pendaftaran berobat
22	END IF
23	IF login = TRUE
24	Redirect halaman home petugas pendaftaran
25	ELSE
26	Sistem menampilkan nomor pendaftaran
27	END IF
28	END IF

Tabel 5.1 Pseudocode Menyimpan Pendaftaran Berobat

29	Selesai
----	---------

Tabel 5.2 berikut menunjukkan perancangan algoritme Membuat Rekam Medis Pasien Baru menggunakan *pseudocode*.

Tabel 5.2 Pseudocode Membuat Rekam Medis Pasien Baru

No	Pseudocode
1	Mulai
2	Input data rekam medis
3	Validasi Form
4	IF form validation = FALSE
5	menampilkan form rekam medis
6	menampilkan pesan kesalahan
7	ELSE
8	Mengambil nilai masukan user
9	Inisialisasi variabel jmlh dengan menghitung baris
10	pada tabel rekam medis ditambah dengan angka 1
11	Inisialisasi nomor rekam medis pasien dengan
12	mengambil karakter pertama pada nama pasien dan
13	ditambah dengan nilai variabel jmlh
14	Inisialisasi nomor member dengan mengambil tanggal
15	dan jam pembuatan rekam medis
16	Insert data ke database
17	IF insert = TRUE
18	Menampilkan pesan "Data berhasil disimpan"
19	ELSE
20	Menampilkan pesan "Data gagal disimpan"
21	END IF
22	Redirect ke halaman index rekam medis pasien
23	END IF
24	Selesai

Tabel 5.3 berikut menunjukkan perancangan algoritme Menambah data rawat jalan gigi menggunakan *pseudocode*.

Tabel 5.3 Pseudocode Menambah Data Rawat Jalan Gigi

No	Pseudocode
1	Mulai
2	Input data rawat jalan gigi
3	Inisialisasi variabel data untuk mengambil nilai masukan
4	dari user
5	Mengetahui nama file yang diunggah yaitu bernama
6	'odontogram'
7	Menentukan penamaan file yang diunggah
8	Inisialisasi variabel config untuk menentukan lokasi
9	penyimpanan file
10	Inisialisasi variabel config untuk menentukan tipe file
11	yang boleh diunggah
12	Inisialisasi variabel config untuk menentukan nama file
13	yang diunggah
14	

Tabel 5.3 Pseudocode Menambah Data Rawat Jalan Gigi

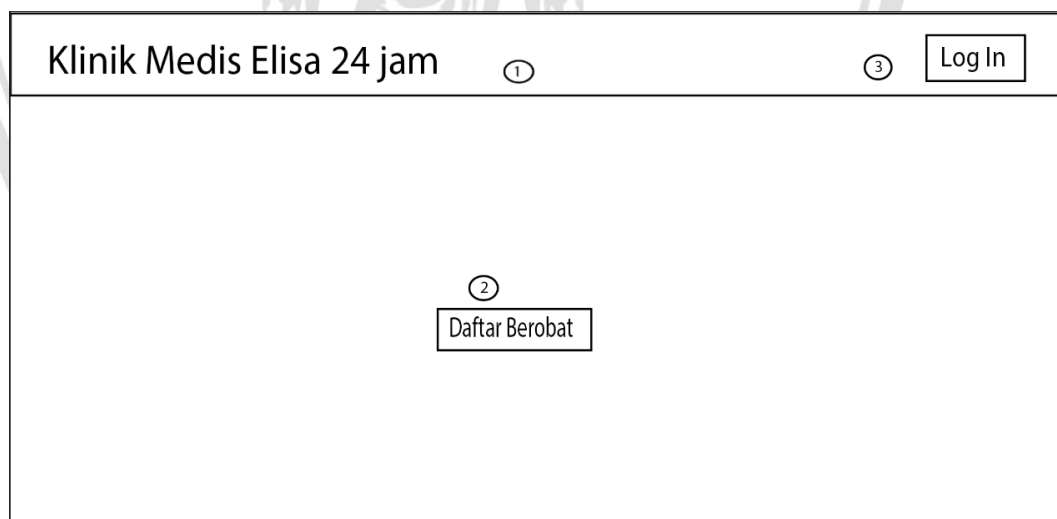
15	Memanggil library upload dengan menggunakan konfigurasi
16	yang telah ditentukan pada variabel config
17	IF file tidak sesuai konfigurasi
18	Redirect form rawat jalan gigi
19	Menampilkan pesan error untuk file unggahan
20	ELSE
21	Insert data ke database
22	IF insert = TRUE
23	Menampilkan pesan "Data berhasil disimpan !"
24	ELSE
25	Menampilkan pesan "Data gagal disimpan !"
26	END IF
27	END IF
28	Redirect halaman indeks rekam medis pasien
	Selesai

5.1.5 Perancangan Antarmuka

Perancangan antarmuka adalah proses untuk membuat antarmuka pada sistem yang dibuat. Fokus pada perancangan antarmuka adalah untuk memaksimalkan kegunaan dari sistem yang dibuat tanpa mengabaikan kenyamanan pengguna dalam mengakses sistem. Perancangan antarmuka dibuat berdasarkan pengalaman sehari-hari (*experience*) *user*. Sehingga sistem yang dihasilkan dapat digunakan dengan mudah oleh *user*.

1. Perancangan antarmuka halaman awal

Gambar 5.6 adalah perancangan antarmuka untuk halaman awal.



Gambar 5.6 Perancangan Antarmuka Halaman Awal

Perancangan antarmuka halaman awal terdiri dari beberapa komponen yaitu seperti dijelaskan pada tabel 5.4 berikut.

Tabel 5.4 Komponen Perancangan Halaman Awal

No	Nama Objek	Keterangan
1	Navbar	Berisi nama perusahaan dan tombol "Log in"
2	Tombol "Daftar Berobat"	Tombol yang akan mengarah ke <i>form</i> daftar berobat
3	Tombol "Log in"	Tombol yang akan mengarah ke halaman <i>log in</i>

2. Perancangan antarmuka halaman *home* petugas pendaftaran

Gambar 5.7 adalah perancangan antarmuka untuk halaman *home* untuk aktor petugas pendaftaran.

Gambar 5.7 Perancangan Antarmuka Halaman *Home* Petugas Pendaftaran

Perancangan antarmuka halaman *home* petugas pendaftaran terdiri dari beberapa komponen yaitu seperti dijelaskan pada tabel 5.5

Tabel 5.5 Komponen Perancangan Halaman *home* Petugas Pendaftaran

No	Nama Objek	Keterangan
1	Navbar	Berisi nama perusahaan dan tombol "Log out"
2	Tombol "Daftar Berobat"	Tombol yang akan mengarah ke <i>form</i> daftar berobat
3	Tombol "Jumlah Kunjungan"	Tombol yang akan mengarah ke halaman yang menampilkan jumlah kunjungan

Tabel 5.5 Komponen Perancangan Halaman *home* Petugas Pendaftaran

4	Tombol “Peringkat Diagnosa”	Tombol yang akan mengarah ke halaman yang menampilkan peringkat diagnosa
5	Tabel Pendaftar	Tabel yang menampilkan data pendaftar
6	Tombol “Log out”	Tombol untuk mengakhiri sesi dan mengarah pada halaman awal
7	Tombol “Konfirmasi”	Tombol untuk mengonfirmasi pendaftaran berobat pasien
8	Tombol “Hapus”	Tombol untuk menghapus data pendaftaran berobat

3. Perancangan antarmuka halaman *home* perawat

Gambar 5.8 adalah perancangan antarmuka untuk halaman *home* untuk aktor Perawat. Halaman ini terdiri dari beberapa komponen yaitu :

Gambar 5.8 Perancangan Antarmuka Halaman *Home* Perawat

Perancangan antarmuka halaman *home* perawat terdiri dari beberapa komponen yaitu seperti dijelaskan pada tabel 5.6.

Tabel 5.6 Komponen Perancangan Halaman *Home* Perawat

No	Nama Objek	Keterangan
1	Navbar	Berisi nama perusahaan dan tombol “Log out”

Tabel 5.6 Komponen Perancangan Halaman *Home* Perawat

2	Tombol “Cari Rekam Medis”	Tombol yang akan mengarah ke halaman pencarian rekam medis
3	Tabel Pendaftar Per poli	Tabel yang menampilkan data pendaftar berdasarkan poli yang dipilih
4	Tombol “Log out”	Tombol untuk mengakhiri sesi dan mengarah pada halaman awal
5	Tombol “Ubah”	Tombol untuk mengubah status pendaftaran berobat pasien

4. Perancangan antarmuka halaman pencarian rekam medis

Gambar 5.9 adalah perancangan antarmuka untuk halaman pencarian rekam medis dengan aktor petugas rekam medis.

Gambar 5.9 Perancangan Antarmuka Halaman Pencarian Rekam Medis

Perancangan antarmuka halaman *home* petugas rekam medis terdiri dari beberapa komponen yaitu seperti dijelaskan pada tabel 5.7

Tabel 5.7 Komponen Perancangan Halaman Pencarian Rekam Medis

No	Nama Objek	Keterangan
1	Navbar	Berisi nama perusahaan dan tombol “Log out”
2	Field “Cari Rekam Medis”	Field untuk mengetikkan nama pasien yang akan dicari rekam medisnya
3	Tombol “Cari Rekam Medis”	Tombol yang akan mengarah ke hasil pencarian rekam medis

Tabel 5.7 Komponen Perancangan Halaman Pencarian Rekam Medis

4	Tombol “Buat RM Baru”	Tombol yang mengarah pada <i>form</i> pembuatan rekam medis baru
5	Tabel Hasil Pencarian Rekam Medis	Tabel yang menampilkan hasil pencarian rekam medis
6	Tombol “Buka”	Tombol untuk membuka rekam medis pada tabel hasil pencarian
7	Tombol “Hapus”	Tombol untuk menghapus rekam medis
8	Tombol “Log out”	Tombol untuk mengakhiri sesi dan mengarah pada halaman awal

5. Perancangan antarmuka halaman *home* petugas laboratorium

Gambar 5.10 adalah perancangan antarmuka untuk halaman *home* untuk aktor petugas laboratorium.

Gambar 5.10 Perancangan Halaman Antarmuka Halaman *Home* Petugas Laboratorium

Perancangan antarmuka halaman *home* petugas laboratorium terdiri dari beberapa komponen yaitu seperti dijelaskan pada tabel 5.8

Tabel 5.8 Komponen Perancangan Halaman *Home* Petugas Laboratorium

No	Nama Objek	Keterangan
1	Navbar	Berisi nama perusahaan dan tombol “log out”

Tabel 5.8 Komponen Perancangan Halaman *Home* Petugas Laboratorium

2	Tabel Daftar Permintaan Diagnostik	Tabel yang menampilkan daftar permintaan pemeriksaan diagnostik
3	Tombol "Detail"	Tombol untuk menampilkan detail permintaan pemeriksaan diagnostik
4	Tombol "Input Hasil"	Tombol untuk memasukkan data hasil pemeriksaan diagnostik
5	Tombol "Log out"	Tombol untuk mengakhiri sesi dan mengarah pada halaman awal

5.2 Implementasi

Implementasi dilakukan dengan mengacu pada perancangan sistem. Implementasi pada penelitian ini terdiri dari spesifikasi pengembangan sistem, implementasi basis data, implementasi algoritme, dan implementasi antarmuka.

5.2.1 Spesifikasi Pengembangan Sistem

Pengembangan sistem ini menggunakan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut.

1. Spesifikasi Perangkat Keras

Processor : Intel(R) Core(TM) i5-6200U
Hard disk drive : 1 TB
RAM : 4096MB
System Model : LENOVO 80Q3

2. Spesifikasi Perangkat Lunak

Operating System : Windows 10 Pro 64-bit (10.0, Build 16299)
Programming Language : PHP
Text Editor : Sublime Text 3.0
Server : XAMPP v3.2.2
Basis data : MariaDB 10.1.30

5.2.2 Implementasi Basis Data

Implementasi basis data dilakukan dengan menggunakan *Data Definition Language (DDL)*. DDL mendefinisikan struktur dari sebuah tabel pada basis data. Implementasi basis data sistem rekam medis elektronik dan pendaftaran pasien *online* pada studi kasus Klinik Medis Elisa dijelaskan pada tabel-tabel berikut.

Tabel 5.9 merupakan implementasi basis data dari tabel admin.

Tabel 5.9 Implementasi Basis Data Tabel admin

No	Kode
1	CREATE TABLE `admin` (
2	`idAdmin` int(11) NOT NULL,
3	`tipe` varchar(20) NOT NULL,
4	`username` varchar(20) NOT NULL,
5	`password` varchar(20) NOT NULL
6) ;
7	
8	ALTER TABLE `admin`
9	ADD PRIMARY KEY (`idAdmin`);
10	
11	ALTER TABLE `admin`
12	MODIFY `idAdmin` int(11) NOT NULL AUTO_INCREMENT,
13	AUTO_INCREMENT=6;
14	COMMIT;

Tabel 5.10 merupakan implementasi basis data dari tabel data pribadi pasien.

Tabel 5.10 Implementasi Basis Data Tabel Data Pribadi Pasien

No	Kode
1	CREATE TABLE `datapribadipasien` (
2	`noRM` varchar(15) NOT NULL,
3	`tanggalDaftar` date NOT NULL,
4	`jenisPembayaran` varchar(20) NOT NULL,
5	`asalPasien` varchar(50) NOT NULL,
6	`namaPasien` varchar(200) NOT NULL,
7	`jenisKelamin` varchar(15) NOT NULL,
8	`statusPerkawinan` varchar(20) NOT NULL,
9	`agama` varchar(20) NOT NULL,
10	`kepercayaan` varchar(20) NOT NULL,
11	`suku` varchar(20) NOT NULL,
12	`bahasa` varchar(20) NOT NULL,
13	`alamatPasien` varchar(200) NOT NULL,
14	`tempatLahir` varchar(50) NOT NULL,
15	`tanggalLahir` date NOT NULL,
16	`noTelpon` varchar(15) NOT NULL,
17	`namaPJPasien` varchar(200) NOT NULL,
18	`alamatPJPasien` varchar(200) NOT NULL,
19	`namaIbu` varchar(200) NOT NULL,
20	`petugasRM` varchar(100) NOT NULL
21);
22	
23	ALTER TABLE `datapribadipasien`
24	ADD PRIMARY KEY (`noRM`);
25	COMMIT;

Tabel 5.11 merupakan implementasi basis data dari tabel diagnostik.

Tabel 5.11 Implementasi Basis Data Tabel Diagnostik

No	Kode
1	CREATE TABLE `datapribadipasien` (
2	`noRM` varchar(15) NOT NULL,

Tabel 5.11 Implementasi Basis Data Tabel Diagnostik

3	`tanggalDaftar` date NOT NULL,
4	`jenisPembayaran` varchar(20) NOT NULL,
5	`asalPasien` varchar(50) NOT NULL,
6	`namaPasien` varchar(200) NOT NULL,
7	`jenisKelamin` varchar(15) NOT NULL,
8	`statusPerkawinan` varchar(20) NOT NULL,
9	`agama` varchar(20) NOT NULL,
10	`kepercayaan` varchar(20) NOT NULL,
11	`suku` varchar(20) NOT NULL,
12	`bahasa` varchar(20) NOT NULL,
13	`alamatPasien` varchar(200) NOT NULL,
14	`tempatLahir` varchar(50) NOT NULL,
15	`tanggalLahir` date NOT NULL,
16	`noTelpon` varchar(15) NOT NULL,
17	`namaPJPasien` varchar(200) NOT NULL,
18	`alamatPJPasien` varchar(200) NOT NULL,
19	`namaIbu` varchar(200) NOT NULL,
20	`petugasRM` varchar(100) NOT NULL
21);
22	
23	ALTER TABLE `datapribadipasien`
24	ADD PRIMARY KEY (`noRM`);
25	COMMIT;

Tabel 5.12 merupakan implementasi basis data dari tabel edukasi rawat jalan.

Tabel 5.12 Implementasi Basis Data Tabel Edukasi Rawat Jalan

1	CREATE TABLE `edukasirawatjalan` (
2	`id` int(11) NOT NULL,
3	`noRM` varchar(20) NOT NULL,
4	`namaPasien` varchar(200) NOT NULL,
5	`tanggalPeriksa` date NOT NULL,
6	`jamPeriksa` time NOT NULL,
7	`materiEdukasi` varchar(225) NOT NULL,
8	`penjelasanDokter` varchar(500) NOT NULL,
9	`pasienMengerti` varchar(10) NOT NULL,
10	`namaKeluarga` varchar(200) NOT NULL,
11	`tenagaMedis` varchar(100) NOT NULL
12);
13	
14	ALTER TABLE `edukasirawatjalan`
15	ADD PRIMARY KEY (`id`);
16	
17	ALTER TABLE `edukasirawatjalan`
18	MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
19	AUTO_INCREMENT=8;
20	COMMIT;

Tabel 5.13 merupakan implementasi basis data dari tabel farmasi.

Tabel 5.13 Implementasi Basis Data Tabel Farmasi

1	CREATE TABLE `farmasi` (
2	`id` int(11) NOT NULL,
3	`tglPermintaan` date NOT NULL,
4	`noRM` varchar(20) NOT NULL,
5	`namaPasien` varchar(200) NOT NULL,
6	`namaObat` varchar(200) NOT NULL,
7	`signa` varchar(100) NOT NULL,
8	`hari` varchar(200) NOT NULL,
9	`jumlahObat` varchar(100) NOT NULL,
10	`dokter` varchar(100) NOT NULL
11);
12	
13	ALTER TABLE `farmasi`
14	ADD PRIMARY KEY (`id`);
15	
16	ALTER TABLE `farmasi`
17	MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
18	COMMIT;

Tabel 5.14 merupakan implementasi basis data dari tabel pendaftaran.

Tabel 5.14 Implementasi Basis Data Tabel Pendaftaran

1	CREATE TABLE `pendaftaran` (
2	`idDaftar` varchar(20) NOT NULL,
3	`tglDaftar` date NOT NULL,
4	`namaPasien` varchar(200) NOT NULL,
5	`umur` int(2) NOT NULL,
6	`jenisAsuransi` varchar(225) NOT NULL,
7	`noKartu` varchar(200) NOT NULL,
8	`poli` varchar(50) NOT NULL,
9	`keperluan` varchar(200) NOT NULL,
10	`noTelp` varchar(12) NOT NULL,
11	`sumber` varchar(50) NOT NULL,
12	`status` varchar(50) NOT NULL DEFAULT 'Belum Dilayani',
13	`statusKonfirmasi` varchar(100) NOT NULL DEFAULT 'Belum
14	Konfirmasi'
15);
16	ALTER TABLE `pendaftaran`
17	ADD PRIMARY KEY (`idDaftar`),
18	ADD UNIQUE KEY `idDaftar` (`idDaftar`);
19	COMMIT;

Tabel 5.15 merupakan implementasi basis data dari tabel pengkajian klinis medis.

Tabel 5.15 Implementasi Basis Data Tabel Pengkajian Klinis Medis

No	Kode
1	CREATE TABLE `pengkajiankm` (
2	`id` int(20) NOT NULL,
3	`noRM` varchar(20) NOT NULL,
4	`namaPasien` varchar(200) NOT NULL,

Tabel 5.15 Implementasi Basis Data Tabel Pengkajian Klinis Medis

5	`tglPemeriksaan` date NOT NULL,
6	`keluhan` varchar(200) NOT NULL,
7	`lamaKeluhan` varchar(100) NOT NULL,
8	`pengobatan` varchar(200) NOT NULL,
9	`riwayatPenyakit` varchar(200) NOT NULL,
10	`riwayatPenyakitklg` varchar(200) NOT NULL,
11	`riwayatAlergi` varchar(200) NOT NULL,
12	`kesadaran` varchar(200) NOT NULL,
13	`tekananDarah` varchar(50) NOT NULL,
14	`denyutJantung` varchar(10) NOT NULL,
15	`suhu` varchar(10) NOT NULL,
16	`pernafasan` varchar(10) NOT NULL,
17	`beratBadan` varchar(10) NOT NULL,
18	`tinggiBadan` varchar(10) NOT NULL,
19	`lingkungan` varchar(100) NOT NULL,
20	`IMT` varchar(100) NOT NULL,
21	`mata` varchar(100) NOT NULL,
22	`telinga` varchar(100) NOT NULL,
23	`hidung` varchar(100) NOT NULL,
24	`tenggorokan` varchar(100) NOT NULL,
25	`leher` varchar(100) NOT NULL,
26	`kelenjarKGB` varchar(100) NOT NULL,
27	`suaraNafas` varchar(100) NOT NULL,
28	`jantung` varchar(100) NOT NULL,
29	`nyeriTekan` varchar(100) NOT NULL,
30	`bisingUsus` varchar(100) NOT NULL,
31	`edema` varchar(10) NOT NULL,
32	`akar` varchar(10) NOT NULL,
33	`gigiBerlubang` varchar(10) NOT NULL,
34	`karangGigi` varchar(10) NOT NULL,
35	`gigiDicabut` varchar(10) NOT NULL,
36	`gusiBerdarah` varchar(10) NOT NULL,
37	`gusiBengkak` varchar(10) NOT NULL,
38	`gigiTambalan` varchar(10) NOT NULL,
39	`gigiKesundulan` varchar(10) NOT NULL,
40	`gigiLain2` varchar(100) NOT NULL,
41	`penunjangLab` varchar(100) NOT NULL,
42	`penunjangRadiologi` varchar(100) NOT NULL,
43	`penunjangLain` varchar(100) NOT NULL,
44	`merokok` varchar(20) NOT NULL,
45	`kurangAktivitas` varchar(10) NOT NULL,
46	`obesitas` varchar(10) NOT NULL,
47	`kontrolTgl` date NOT NULL,
48	`kontrolJam` time NOT NULL,
49	`kontrolTempat` varchar(100) NOT NULL,
50	`diagnosa` varchar(200) NOT NULL,
51	`pemecahanMasalah` varchar(200) NOT NULL,
52	`petugas` varchar(100) NOT NULL
53);
54	
55	ALTER TABLE `pengkajiankm`
56	ADD PRIMARY KEY (`id`);
57	
58	ALTER TABLE `pengkajiankm`
59	MODIFY `id` int(20) NOT NULL AUTO_INCREMENT,
60	AUTO_INCREMENT=7;
61	COMMIT;

Tabel 5.16 merupakan implementasi basis data dari tabel rawat jalan.

Tabel 5.16 Implementasi Basis Data Tabel Rawat Jalan

No	Kode
1	CREATE TABLE `rawatjalan` (
2	`id` int(11) NOT NULL,
3	`noRM` varchar(20) NOT NULL,
4	`namaPasien` varchar(200) NOT NULL,
5	`jenis` varchar(20) NOT NULL,
6	`tglPemeriksaan` date NOT NULL,
7	`jamPemeriksaan` time NOT NULL,
8	`keluhan` varchar(200) NOT NULL,
9	`terapi` varchar(200) NOT NULL,
10	`diagnosa` varchar(200) NOT NULL,
11	`diagnosaSekunder` varchar(200) NOT NULL,
12	`tenagaMedis` varchar(200) NOT NULL,
13	`statusPulang` varchar(100) NOT NULL,
14	`odontogram` varchar(225) DEFAULT NULL
15);
16	
17	ALTER TABLE `rawatjalan`
18	ADD PRIMARY KEY (`id`);
19	
20	ALTER TABLE `rawatjalan`
21	MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
22	AUTO_INCREMENT=15;
23	COMMIT;

5.2.3 Implementasi Algoritme

1. Implementasi Algoritme Menyimpan Pendaftaran Berobat

Implementasi dari Algoritme Menyimpan Pendaftaran Berobat terdapat pada fungsi `savedaftarBerobat()` yang terletak pada kelas *controller* pendaftaran. Fungsi ini memanggil *form_validation* untuk memeriksa masukan *user* pada *form* daftar. Jika salah maka akan menampilkan kembali *form* daftar dan pesan *error*. Jika masukan sudah benar maka akan melakukan inisialisasi `idDaftar`, tanggal daftar, dan inisialisasi sumber. Lalu akan menampilkan nomor pendaftaran dan melakukan *insert* data ke database. Jika transaksi berhasil akan menampilkan pesan “Data berhasil disimpan”. Jika gagal, akan menampilkan “data gagal disimpan”. Kemudian sistem akan memeriksa apakah *user* sudah *log in*. Jika benar akan *redirect* ke halaman *home* petugas pendaftaran. Jika salah (yang berarti *user* adalah *guest*), sistem menampilkan nomor pendaftaran.

Nama fungsi : `savedaftarBerobat()`

Nama kelas : `pendaftaran`

Kode program :

Tabel 5.17 Implementasi Algoritme Menyimpan Pendaftaran Berobat

No	Kode
1	public function savedaftarBerobat()
2	{
3	
4	if (\$this->form_validation->run('formdaftar') ==
5	FALSE) {
6	\$this->load->view('formdaftar');
7	}else{
8	\$data = \$this->input->post();
9	\$data['idDaftar'] = date('mdHis').(\$this->
10	getRandomWord());
11	\$data['tglDaftar'] = date('Y-m-d');
12	if (\$this->session->userdata('login')) {
13	\$data['sumber'] = "datang langsung";
14	}else{
15	\$data['sumber'] = "aplikasi";
16	}
17	\$insert = \$this->m_pendaftaran->
18	savedaftarBerobat(\$data);
19	if (\$insert) {
20	print "<script
21	type=\"text/javascript\">alert('Data Berhasil
22	Disimpan');</script>";
23	}else{
24	print "<script
25	type=\"text/javascript\">alert('Data Gagal
26	Disimpan');</script>";
27	redirect('pendaftaran/daftarBerobat',
28	'refresh');
29	}
30	if (\$this->session->userdata('login')) {
31	redirect('pendaftaran/allPendaftaran');
32	}else{
33	\$this->load->view('daftarBerhasil', \$data);
34	}
35	}

2. Implementasi Algoritme Membuat Rekam Medis Baru

Implementasi dari Algoritme Daftar Berobat terdapat pada fungsi `saveRMBaru()` yang terletak pada kelas *controller* rekam medis. Setelah data dimasukkan oleh petugas rekam medis pada *form* rekam medis baru, maka akan melakukan validasi *form*. Jika salah maka akan menampilkan kembali *form* daftar dan pesan *error*. Jika benar, akan mengambil masukan petugas rekam medis menggunakan method POST. Kemudian dilakukan proses memasukkan data ke dalam *database* melalui fungsi `saveRMBaru()` pada model `m_rekam medis`. Jika transaksi berhasil, akan menampilkan pesan bahwa data berhasil disimpan. Jika transaksi gagal, akan menampilkan pesan data gagal disimpan.

Nama fungsi : `saveRMBaru()`

Nama kelas : `rekam medis`

Kode program :

Tabel 5.18 Implementasi Algoritme Membuat Rekam Medis Baru

No	Kode
1	public function saveRMBaru()
2	{
3	if (\$this->form_validation-
4	>run('formrekammedisbaru') == false) {
5	\$this->load->view('form_input/formRMBaru');
6	} else {
7	\$data = \$this->input->post();
8	\$jmlh = \$this->m_rekammedis-
9	>countNoRM(substr(\$data['namaPasien'], 0, 1))+1;
10	\$data['noRM'] = substr(\$data['namaPasien'], 0,
11	1).\$jmlh
12	\$data['noMember'] = date('ymdHi');
13	\$insert = \$this->m_rekammedis-
14	>saveRMBaru(\$data);
15	if (\$insert) {
16	print "<script
17	type=\"text/javascript\">alert('Data Berhasil
18	Disimpan');</script>";
19	} else {
20	print "<script
21	type=\"text/javascript\">alert('Data Gagal
22	Disimpan');</script>";
23	} //end else insert
24	redirect('index/' . \$data['noRM'],
25	'refresh');
26	}
27	}

3. Implementasi Algoritme Menambah Data Rawat Jalan Gigi

Implementasi dari Algoritme Menambah Data Rawat Jalan Gigi terdapat pada fungsi `saveRawatJalanGigi()` yang terletak pada kelas *controller* `rawatjalan`. Setelah pengguna memasukkan data pada *form* data rawat jalan gigi, maka data pada *form* akan diambil menggunakan *method post*. Lalu akan dilakukan inisialisasi nama *file* dari foto odontogram yang telah diunggah dan akan disimpan pada kolom odontogram. Kemudian dilakukan konfigurasi untuk menentukan lokasi *file*, tipe *file* yang boleh diunggah dan nama *file*. Jika *file* yang diunggah belum memenuhi konfigurasi yang sudah ditentukan, maka sistem akan menampilkan peringatan bahwa data gagal disimpan. Jika *file* sudah sesuai ketentuan, maka data berhasil disimpan.

Nama fungsi : `saveRawatJalanGigi()`

Nama kelas : `rawatjalan`

Kode program :

Tabel 5.19 Implementasi Algoritme Menambah data rawat jalan gigi

No	Kode
1	public function saveRawatJalanGigi()
2	{
3	\$data = \$this->input->post();
4	\$filename = \$_FILES['odontogram']['name'];
5	\$data['odontogram'] =
6	'odontogram_' . date('dmYHis') . '.' . substr(\$filename, strrpos(
7	\$filename, '.')+1);
8	\$config['upload_path'] = './uploads/';
9	\$config['allowed_types'] = 'gif jpg png';
10	\$config['file_name'] =
11	\$data['odontogram'];
12	
13	\$this->load->library('upload', \$config);
14	
15	if (!\$this->upload->do_upload('odontogram')){
16	\$gambar = array('error' => \$this->upload->
17	display_errors());
18	} else {
19	\$insert= \$this->m_rekammedis->
20	saveRawatJalan(\$data);
21	if (\$insert) {
22	print "<script
23	type=\"text/javascript\">alert('Data Berhasil
24	Disimpan');</script>";
25	}
26	else{
27	print "<script
28	type=\"text/javascript\">alert('Data Gagal
29	Disimpan');</script>";
30	
31	}
32	}
33	redirect('rekammedis/index/' . \$data['noRM'], 'refresh');
34	
35	}

5.2.4 Implementasi Antarmuka

1. Implementasi Halaman Awal

Halaman awal terdiri dari *navbar*, tombol “Daftar Berobat”, Tombol “Log in”, dan tombol “Contact” yang jika ditekan akan memberikan informasi alamat, nomor telpon, dan alamat *e-mail* Klinik Medis Elisa.

Implementasi Halaman Awal ditunjukkan pada gambar 5.11.



Gambar 5.11 Implementasi Halaman Awal

2. Implementasi Halaman *Home* Petugas Pendaftaran

Halaman *Home* Petugas Pendaftaran terdiri dari navbar, tombol “Daftar Berobat”, tombol “Jumlah Kunjungan”, tombol “Peringkat Diagnosa”, tabel data pendaftar, tombol “Konfirmasi”, tombol “Hapus”, dan tombol “Log out”. Implementasi Halaman *Home* Petugas Pendaftaran ditampilkan pada halaman 5.12.

No.	ID Pendaftaran.	Tanggal	Nama Pasien	Umur	Poli	Keperluan	No Telp	Sumber	Status Konfirmasi	Action	Action
1	0919084755ABC	2018-09-19	Sonya Maharani	20	gigi	kontrol behel	08123456790	aplikasi	Belum Konfirmasi	Konfirmasi	Hapus
2	0919084912ABC	2018-09-19	Grace Anglea	20	gigi	periksa	08123333333	datang langsung	Belum Konfirmasi	Konfirmasi	Hapus
3	0919085001ABC	2018-09-19	Clara Pusparani	20	umum	kontrol behel	08123456790	datang langsung	Belum Konfirmasi	Konfirmasi	Hapus
4	091808704ABC	2018-09-18	Clara Pusparani	22	gigi	periksa	08123456790	datang langsung	Sudah Konfirmasi	Konfirmasi	Hapus
5	0828032523ABC	2018-08-28	Sonya Maharani	20	gigi	periksa	08123333333	aplikasi	Sudah Konfirmasi	Konfirmasi	Hapus
6	0828033634ABC	2018-08-28	Grace Anglea	22	umum	periksa	08123333333	datang langsung	Sudah Konfirmasi	Konfirmasi	Hapus
7	0827055532ABC	2018-08-27	Sonya Maharani	20	gigi	periksa	08123456790	aplikasi	Sudah Konfirmasi	Konfirmasi	Hapus
8	0823201222ABC	2018-08-23	Clara Pusparani	20	umum	berobat	08123456790	aplikasi	Sudah Konfirmasi	Konfirmasi	Hapus

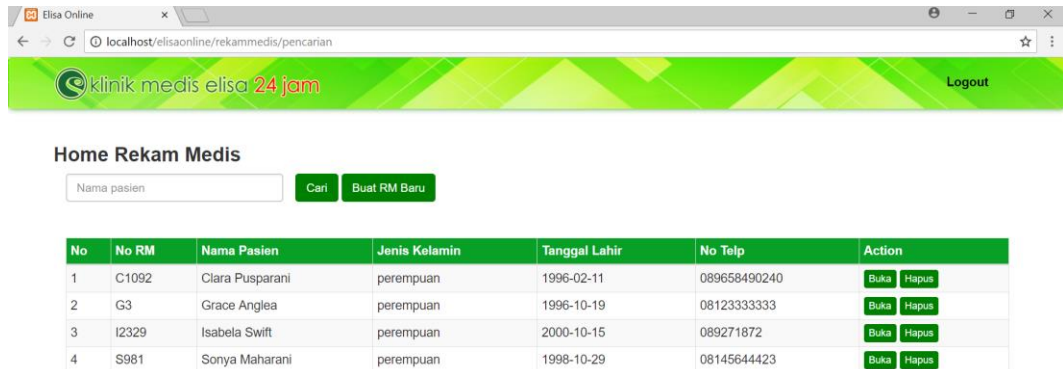
Gambar 5.12 Implementasi Halaman *Home* Petugas Pendaftaran

3. Implementasi Halaman Pencarian Rekam Medis

Halaman Pencarian Rekam Medis terdiri dari navbar, kolom pencarian rekam medis, tombol “Buat RM Baru”, tombol “Cari”, tabel yang menampilkan hasil

pencarian rekam medis, tombol “Buka”, tombol “Hapus”, dan tombol “Log out”.

Implementasi Halaman Pencarian Rekam Medis ditampilkan pada gambar 5.13

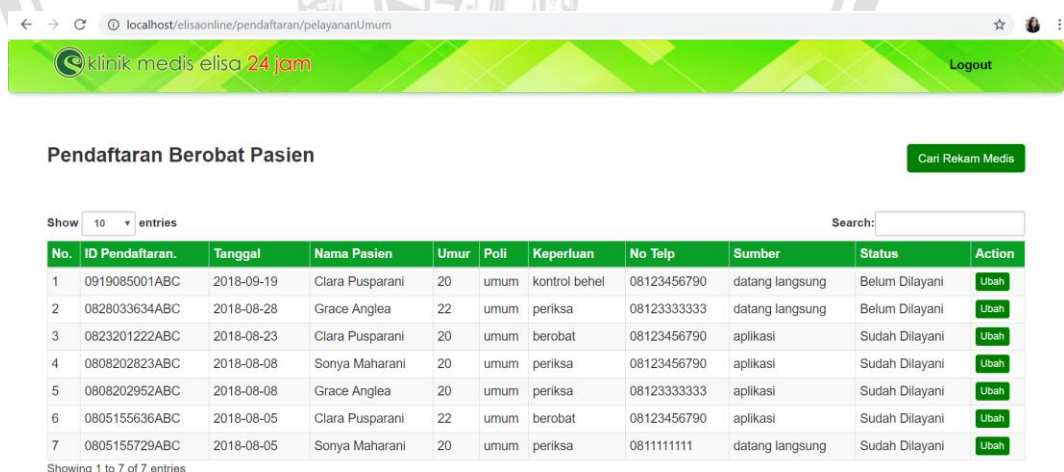


Gambar 5.13 Implementasi Halaman Pencarian Rekam Medis

4. Implementasi Halaman *Home* Perawat

Halaman *Home* Petugas Perawat terdiri dari navbar, tombol “Cari Rekam Medis”, tabel yang menampilkan data pendaftar, tombol untuk mengubah status antrian pendaftaran, dan tombol “Log out”.

Implementasi Halaman *Home* Perawat dapat dilihat pada gambar 5.14.



Gambar 5.14 Implementasi Halaman *Home* Perawat

5. Implementasi Halaman *Home* Petugas Laboratorium

Halaman Home Petugas Laboratorium terdiri dari navbar, tabel yang menampilkan data permintaan pemeriksaan diagnostik, tombol untuk melihat detail permintaan pemeriksaan diagnostik, tombol untuk memasukkan hasil pemeriksaan diagnostik, dan tombol “Log out”. Implementasi Halaman Home Petugas Laboratorium dapat dilihat pada gambar 5.15.



The screenshot shows a web application interface for a laboratory. At the top, there is a green header with the logo 'klinik medis elisa 24 jam' and a 'Logout' button. Below the header, the title 'Permintaan Pemeriksaan Diagnostik' is displayed. A table lists four diagnostic test requests. Each row includes a number, date, patient name, age, test type, referring doctor, status, and two action buttons: 'Detail' and 'Input Hasil'.

No.	Tanggal Permintaan	Nama Pasien	Umur	Jenis Pemeriksaan	Dokter Pengirim	Status	Action	Action
1	2018-08-28	Clara Pusparani	20 tahun	Darah Lengkap	dr.Vebby	Sudah Dilayani	Detail	Input Hasil
2	2018-08-15	Clara Pusparani	20 tahun	serologi	dr.Vebby	Sudah Dilayani	Detail	Input Hasil
3	2018-08-09	Clara Pusparani	20 tahun	serologi	dr.Vebby	Sudah Dilayani	Detail	Input Hasil
4	2018-08-09	Sonya Maharani	20 tahun	darah lengkap, widal	dr.Vebby	Sudah Dilayani	Detail	Input Hasil

Showing 1 to 4 of 4 entries

Gambar 5.15 Implementasi Halaman *Home* Petugas Laboratorium

BAB 6 PENGUJIAN

6.1 Pengujian Unit

Pada level pengujian unit, akan dilakukan pengujian pada setiap unit atau komponen pada sebuah perangkat lunak. Tujuan pengujian ini adalah untuk memastikan bahwa masing-masing unit telah bekerja sesuai dengan perancangan. Pengujian unit dilakukan dengan menggunakan *Basis Path Testing* yang menguji setiap jalur yang ada pada sebuah unit.

Pengujian menggunakan *basis path testing* dilakukan dengan menentukan *flow graph* berdasarkan dari algoritme dari fungsi utama. Kemudian dilakukan penentuan jalur independen dan penghitungan nilai *cyclomatic complexity* sehingga dapat menentukan kasus uji dan hasil uji.

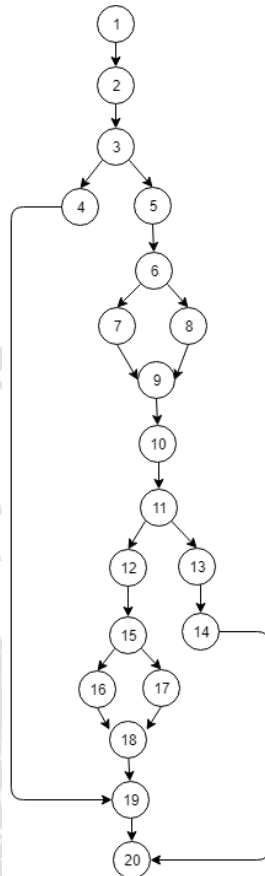
6.1.1 Pengujian unit Fungsi Menyimpan Pendaftaran Berobat

Pseudocode Menyimpan Pendaftaran Berobat ditunjukkan pada Tabel 6.1.

Tabel 6.1 Pseudocode Menyimpan Pendaftaran Berobat

Pseudocode	Node
Mulai	1
Input data pendaftaran	2
Validasi Form	3
IF form validation = FALSE	4
menampilkan form pendaftaran	
menampilkan pesan kesalahan	
ELSE	5
Mengambil nilai masukan user	
Inisialisasi nomor pendaftaran	
Inisialisasi tanggal pendaftaran	
IF login = TRUE	6
Inisialisasi sumber = "datang langsung"	7
ELSE	8
Inisialisasi sumber = "aplikasi"	
END IF	9
Insert data ke database	10
IF insert = TRUE	11
Menampilkan pesan "Data berhasil disimpan"	12
ELSE	13
Menampilkan pesan "Data gagal disimpan"	
Redirect halaman pendaftaran berobat	
END IF	14
IF login = TRUE	15
Redirect halaman home petugas pendaftaran	16
ELSE	17
Sistem menampilkan nomor pendaftaran	
END IF	18
END IF	19
Selesai	20

Berdasarkan *pseudocode* Menyimpan Pendaftaran Berobat, maka dapat dihasilkan *flow graph* pada gambar 6.1.



Gambar 6.1 *Flow Graph* Daftar Berobat Pasien Lama

1. Jalur independen (*Independent Path*):

Jalur 1 : 1-2-3-4-19-20

Jalur 2 : 1-2-3-5-6-7-9-10-11-13-14-20

Jalur 3 : 1-2-3-5-6-8-9-10-11-13-14-20

Jalur 4 : 1-2-3-5-6-7-9-10-11-12-15-16-18-19-20

Jalur 5 : 1-2-3-5-6-8-9-10-11-12-15-17-18-19-20

2. Perhitungan *cyclomatic complexity*:

$$V(G) = \text{Jumlah edge} - \text{Jumlah node} + 2$$

$$= 23 - 20 + 2$$

$$= 5$$

$$V(G) = \text{Jumlah Predicate Node} + 1$$

$$= 4 + 1$$

$$= 5$$

$V(G)$ = Jumlah region

= 5

Berdasarkan jalur independen yang didapatkan, maka dilakukan pengujian dengan kasus uji seperti pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Unit Fungsi Menyimpan Pendaftaran Berobat

Jalur	Prosedur Uji	Expected Result	Result	Status
1	Nilai form_validation = FALSE, di mana ada <i>field</i> pada <i>form</i> pendaftaran berobat yang belum diisi	Sistem akan menjalankan <i>statement</i> untuk hasil IF form_validation = FALSE yaitu menampilkan kembali <i>form</i> pendaftaran berobat beserta pesan kesalahan pada <i>field</i> yang belum diisi	Sistem berhasil menjalankan <i>statement</i> untuk hasil IF form_validation = FALSE yaitu menampilkan kembali <i>form</i> pendaftaran berobat beserta pesan kesalahan pada <i>field</i> yang belum diisi	valid
2	<ol style="list-style-type: none"> 1. Nilai form_validation = TRUE 2. Nilai login = TRUE, di mana yang melakukan pendaftaran adalah petugas pendaftaran 3. Nilai insert = FALSE. Di mana koneksi dengan basis data terputus saat melakukan penyimpanan data 	<ol style="list-style-type: none"> 1. Kolom "sumber" pada tabel pendaftaran akan memiliki nilai "datang langsung". 2. Sistem akan menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data 	<ol style="list-style-type: none"> 1. Kolom "sumber" pada tabel pendaftaran memiliki nilai "datang langsung". 2. Sistem berhasil menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data 	valid
3	<ol style="list-style-type: none"> 1. Nilai form_validation = TRUE 2. Nilai login = FALSE, di mana yang melakukan pendaftaran adalah <i>guest</i> 3. Nilai insert = FALSE, di mana koneksi dengan basis data terputus saat melakukan penyimpanan data 	<ol style="list-style-type: none"> 1. Kolom "sumber" pada tabel pendaftaran akan memiliki nilai "aplikasi". 2. Sistem akan menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data 	<ol style="list-style-type: none"> 1. Kolom "sumber" pada tabel pendaftaran memiliki nilai "aplikasi". 2. Sistem berhasil menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data 	valid
4	<ol style="list-style-type: none"> 1. Nilai form_validation = TRUE 2. Nilai login = TRUE, di mana yang melakukan 	<ol style="list-style-type: none"> 1. Kolom "sumber" pada tabel pendaftaran akan memiliki nilai 	<ol style="list-style-type: none"> 1. Kolom "sumber" pada tabel pendaftaran memiliki nilai 	valid

	<p>pendaftaran adalah petugas pendaftaran</p> <p>3. Nilai insert = TRUE</p>	<p>“datang langsung”</p> <p>2. Sistem akan menampilkan pesan bahwa data berhasil disimpan</p> <p>3. Setelah menyimpan data, sistem akan <i>redirect</i> ke halaman utama petugas pendaftaran</p>	<p>“datang langsung”</p> <p>2. Sistem berhasil menampilkan pesan bahwa data berhasil disimpan</p> <p>3. Setelah menyimpan data, sistem berhasil <i>redirect</i> ke halaman utama petugas pendaftaran</p>	
5	<p>1. Nilai form_validation = TRUE</p> <p>2. Nilai login = FALSE, di mana yang melakukan pendaftaran adalah <i>guest</i></p> <p>3. Nilai insert = TRUE</p>	<p>1. Kolom “sumber” pada tabel pendaftaran akan memiliki nilai “aplikasi”</p> <p>2. Sistem akan menampilkan pesan bahwa data berhasil disimpan</p> <p>3. Setelah menyimpan data, sistem akan <i>redirect</i> ke halaman yang menampilkan nomor pendaftaran pasien</p>	<p>1. Kolom “sumber” pada tabel pendaftaran memiliki nilai “aplikasi”</p> <p>2. Sistem berhasil menampilkan pesan bahwa data berhasil disimpan</p> <p>3. Setelah menyimpan data, sistem berhasil <i>redirect</i> ke halaman yang menampilkan nomor pendaftaran pasien</p>	valid

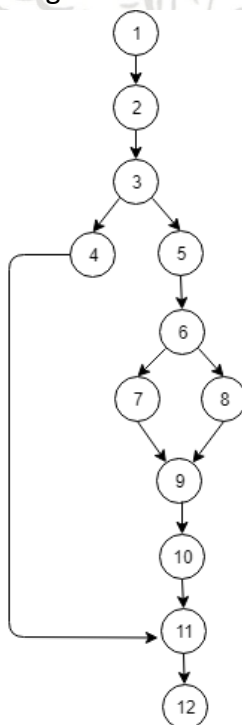
6.1.2 Pengujian unit Fungsi Membuat Rekam Medis Baru

Pseudocode Membuat Rekam Medis Baru ditunjukkan pada Tabel 6.3.

Tabel 6.3 Pseudocode Membuat Rekam Medis Baru

Pseudocode	Node
Mulai	1
Input data rekam medis	2
Validasi Form	3
IF form validation = FALSE	4
menampilkan form rekam medis	
menampilkan pesan kesalahan	
ELSE	5
Mengambil nilai masukan user	
Inisialisasi variabel jmlh dengan menghitung baris pada tabel rekam medis ditambah dengan angka 1	
Inisialisasi nomor rekam medis pasien dengan mengambil karakter pertama pada nama pasien dan ditambah dengan nilai variabel jmlh	
Inisialisasi nomor member dengan mengambil tanggal dan jam pembuatan rekam medis	
Insert data ke database	6
IF insert = TRUE	7
Menampilkan pesan "Data berhasil disimpan"	8
ELSE	9
Menampilkan pesan "Data gagal disimpan"	
END IF	10
Redirect ke halaman index rekam medis pasien	11
END IF	
Selesai	12

Berdasarkan *pseudocode* Membuat Rekam Medis Baru, maka dapat dihasilkan *flow graph* pada gambar 6.2.



Gambar 6.2 Flow Graph Membuat Rekam Medis Baru

1. Jalur independen (*Independent Path*):

I : 1-2-3-4-11-12

II : 1-2-3-5-6-7-9-10-11-12

III : 1-2-3-4-6-8-9-10-11-12

2. Perhitungan *cyclomatic complexity*:

$V(G) = \text{Jumlah edge} - \text{Jumlah node} + 2$

$= 13 - 12 + 2$

$= 3$

$V(G) = \text{Jumlah predicate node} + 1$

$= 2 + 1$

$= 3$

$V(G) = \text{Jumlah region}$

$= 3$

Berdasarkan jalur independen yang didapatkan, maka dilakukan pengujian dengan kasus uji seperti pada Tabel 6.4.

Tabel 6.4 Kasus Uji Membuat Rekam Medis Baru

Jalur	Prosedur Uji	<i>Expected Result</i>	<i>Result</i>	Status
1	Nilai <i>form_validation</i> = FALSE, di mana ada <i>field</i> pada <i>form</i> rekam medis baru yang belum diisi	Sistem akan menjalankan <i>statement</i> untuk hasil IF <i>form_validation</i> = FALSE yaitu menampilkan kembali <i>form</i> rekam medis baru beserta pesan kesalahan pada <i>field</i> yang belum diisi	Sistem berhasil menjalankan <i>statement</i> untuk hasil IF <i>form_validation</i> = FALSE yaitu menampilkan kembali <i>form</i> rekam medis baru beserta pesan kesalahan pada <i>field</i> yang belum diisi	valid
2	1. Nilai <i>form_validation</i> = TRUE 2. Nilai insert = TRUE	1. Sistem akan menampilkan pesan bahwa data berhasil disimpan 2. Setelah menyimpan data, sistem akan <i>redirect</i> ke halaman rekam medis pada <i>tab</i>	1. Sistem berhasil menampilkan pesan bahwa data berhasil disimpan 2. Setelah menyimpan data, sistem berhasil <i>redirect</i> ke halaman rekam medis	valid

		Data Pribadi Pasien	pada <i>tab</i> Data Pribadi Pasien	
3	<ol style="list-style-type: none"> 1. Nilai form_validation = TRUE 2. Nilai insert = FALSE, di mana koneksi dengan basis data terputus saat melakukan penyimpanan data 	Sistem akan menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data	Sistem berhasil menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data	valid

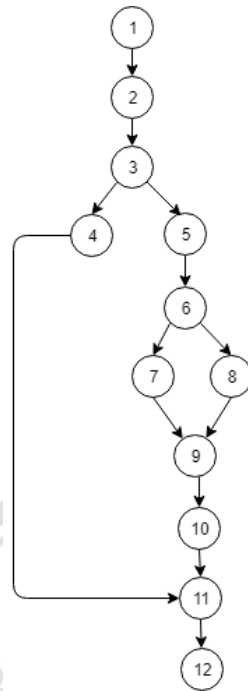
6.1.3 Pengujian unit Fungsi Menambah Data Rawat Jalan Gigi

Pseudocode Menambah Data Rawat Jalan Gigi ditunjukkan pada tabel 6.5.

Tabel 6.5 *Pseudocode* Menambah Data Rawat Jalan Gigi

Pseudocode	Node
Mulai	1
Input data rawat jalan gigi	
Inisialisasi variabel data untuk mengambil nilai masukan dari user	
Mengetahui nama file yang diunggah yaitu bernama 'odontogram'	
Menentukan penamaan file yang diunggah	
Inisialisasi variabel config untuk menentukan lokasi penyimpanan file	2
Inisialisasi variabel config untuk menentukan tipe file yang boleh diunggah	
Inisialisasi variabel config untuk menentukan nama file yang diunggah	
Memanggil library upload dengan menggunakan konfigurasi yang telah ditentukan pada variabel config	
IF file tidak sesuai konfigurasi	3
Redirect form rawat jalan gigi	
Menampilkan pesan error untuk file unggahan	4
ELSE	5
Insert data ke database	
IF insert = TRUE	6
Menampilkan pesan "Data berhasil disimpan !"	7
ELSE	8
Menampilkan pesan "Data gagal disimpan !"	
END IF	9
END IF	10
Redirect halaman indeks rekam medis pasien	11
Selesai	12

Berdasarkan algoritme Menambah Data Rawat Jalan Gigi, maka dapat dihasilkan *flow graph* pada gambar 6.3.



Gambar 6.3 Flow Graph Menambah Data Rawat Jalan Gigi

1. Jalur independen (*Independent Path*):

I : 1-2-3-4-11-12

II : 1-2-3-5-6-7-9-10-11-12

III : 1-2-3-5-6-8-9-10-11-12

2. Perhitungan *cyclomatic complexity*:

$$\begin{aligned}
 V(G) &= \text{Jumlah edge} - \text{Jumlah node} + 2 \\
 &= 13 - 12 + 2 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= \text{Jumlah predicate node} + 1 \\
 &= 2 + 1 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= \text{Jumlah region} \\
 &= 3
 \end{aligned}$$

Berdasarkan jalur independen yang didapatkan, maka dilakukan pengujian dengan kasus uji seperti pada tabel 6.6 berikut.

Tabel 6.6 Kasus Uji Menambah Data Rawat Jalan Gigi

Jalur	Prosedur Uji	Expected Result	Result	Status
1	Nilai <i>file</i> != konfigurasi, di mana format <i>file</i> foto unggahan salah	Sistem akan menjalankan <i>statement</i> untuk hasil <i>file</i> != konfigurasi, yaitu menampilkan kembali <i>form</i> rawat jalan gigi beserta pesan kesalahan mengunggah <i>file</i> foto	Sistem berhasil menjalankan <i>statement</i> untuk hasil <i>file</i> != konfigurasi, yaitu menampilkan kembali <i>form</i> rawat jalan gigi beserta pesan kesalahan mengunggah <i>file</i> foto	valid
2	1. Nilai <i>file</i> = konfigurasi 2. Nilai insert = TRUE	1. Sistem akan menampilkan pesan bahwa data berhasil disimpan 2. Setelah menyimpan data, sistem akan <i>redirect</i> ke halaman rekam medis pasien	1. Sistem berhasil menampilkan pesan bahwa data berhasil disimpan 2. Setelah menyimpan data, sistem berhasil <i>redirect</i> ke halaman rekam medis pasien	valid
3	1. Nilai <i>file</i> = konfigurasi 2. Nilai insert = FALSE, di mana koneksi dengan basis data terputus saat melakukan penyimpanan data	Sistem akan menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data	Sistem berhasil menampilkan pesan kesalahan koneksi basis data dan gagal menyimpan data	valid

6.2 Pengujian Validasi

Pengujian validasi dilakukan menggunakan metode *black box* dengan jenis *scenario-based testing*. Pada pengujian ini, penguji menjalankan *use case* berikut semua alternatif yang terdapat didalamnya. Tujuan pengujian ini adalah menemukan adanya kesalahan pada saat pengguna berinteraksi dengan sistem.

Kasus uji daftar berobat ditunjukkan pada Tabel 6.7 berikut.

Tabel 6.7 Kasus Uji Menyimpan Pendaftaran Berobat

Nama Kasus Uji	Kasus Uji Menyimpan Pendaftaran Berobat
Kode Kebutuhan	RME_F_001
Prosedur	1. Aktor memilih menu “Daftar Berobat” 2. Aktor mengisi <i>form</i> pendaftaran berobat secara lengkap 3. Aktor menekan tombol “Simpan”

Tabel 6.7 Kasus Uji Menyimpan Pendaftaran Berobat

Hasil yang diharapkan	<ol style="list-style-type: none"> 1. Sistem berhasil menyimpan data ke <i>database</i>. Untuk <i>guest</i>, sistem menampilkan halaman baru yang berisi notifikasi bahwa pendaftaran berhasil dan nomor pendaftaran 2. Sistem berhasil menyimpan data ke <i>database</i>. Untuk petugas pendaftaran, sistem akan kembali menampilkan halaman utama petugas pendaftaran yang berisi tabel pendaftaran pasien yang telah diperbaharui.
Hasil	<ol style="list-style-type: none"> 1. Sistem berhasil menyimpan data ke <i>database</i>. Untuk <i>guest</i>, sistem menampilkan halaman baru yang berisi notifikasi bahwa pendaftaran berhasil dan nomor pendaftaran 2. Sistem berhasil menyimpan data ke <i>database</i>. Untuk petugas pendaftaran, sistem akan kembali menampilkan halaman utama petugas pendaftaran yang berisi tabel pendaftaran pasien yang telah diperbaharui.
Status	Valid

Kasus uji Menyimpan Pendaftaran Berobat alternatif 1 ditunjukkan pada Tabel 6.8 berikut.

Tabel 6.8 Kasus Uji Menyimpan Pendaftaran Berobat Alternatif 1 : Validasi *form* gagal

Nama Kasus Uji	Kasus Uji Daftar Berobat
Kode Kebutuhan	RME_F_001
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu “Daftar Berobat” 2. Aktor mengisi <i>form</i> pendaftaran berobat dengan mengosongkan satu <i>field</i> 3. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem kembali pada halaman <i>form</i> pendaftaran dan menampilkan pesan kesalahan berupa “Field ...tidak boleh kosong” pada <i>field</i> yang belum diisi
Hasil	Sistem kembali pada halaman <i>form</i> pendaftaran dan menampilkan pesan kesalahan berupa “Field ...tidak boleh kosong” pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji Menyimpan Pendaftaran Berobat alternatif 2 ditunjukkan pada Tabel 6.9 berikut.

Tabel 6.9 Kasus Uji Menyimpan Pendaftaran Berobat Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Daftar Berobat
Kode Kebutuhan	RME_F_001

Tabel 6.9 Kasus Uji Menyimpan Pendaftaran Berobat Alternatif 2 : Koneksi dengan basis data terputus

Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu “Daftar Berobat” 2. Aktor mengisi <i>form</i> pendaftaran berobat secara lengkap 3. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji *Log in* ditunjukkan pada Tabel 6.10 berikut.

Tabel 6.10 Kasus Uji *Log in*

Nama Kasus Uji	Kasus Uji <i>Log in</i>
Kode Kebutuhan	RME_F_002
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu “<i>Log in</i>” 2. Aktor mengisi semua <i>field username</i> dan <i>password</i> 3. Aktor menekan tombol “<i>Log in</i>”
Hasil yang diharapkan	Sistem menampilkan halaman utama masing-masing pengguna berdasarkan tipe
Hasil	Sistem menampilkan halaman utama masing-masing pengguna berdasarkan tipe
Status	Valid

Kasus uji *Log in* alternatif 1 ditunjukkan pada Tabel 6.11 berikut.

Tabel 6.11 Kasus Uji *Log in* Alternatif 1 : Validasi *form* gagal

Nama Kasus Uji	Kasus Uji <i>Log in</i>
Kode Kebutuhan	RME_F_002
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu “<i>Log in</i>” 2. Aktor mengisi semua <i>field username</i> dan <i>password</i> 3. Aktor menekan tombol “<i>Log in</i>”
Hasil yang diharapkan	Sistem kembali pada halaman <i>form</i> pendaftaran dan menampilkan pesan kesalahan berupa “Field ...tidak boleh kosong” pada <i>field</i> yang belum diisi
Hasil	Sistem kembali pada halaman <i>form</i> pendaftaran dan menampilkan pesan kesalahan berupa “Field ...tidak boleh kosong” pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji *Log in* alternatif 2 ditunjukkan pada Tabel 6.12 berikut.

Tabel 6.12 Kasus Uji *Log in* Alternatif 2 : *Username* atau *password* salah

Nama Kasus Uji	Kasus Uji <i>Log in</i>
Kode Kebutuhan	RME_F_002
Prosedur	1. Aktor memilih menu " <i>Log in</i> " 2. Aktor mengisi <i>field username</i> dan <i>password</i> 3. Aktor menekan tombol " <i>Log in</i> "
Hasil yang diharapkan	Sistem menampilkan peringatan bahwa kombinasi <i>username</i> dan <i>password</i> salah dan kembali pada halaman <i>log in</i>
Hasil	Sistem menampilkan peringatan bahwa kombinasi <i>username</i> dan <i>password</i> salah dan kembali pada halaman <i>log in</i>
Status	Valid

Kasus uji melihat *list* pendaftaran berobat ditunjukkan pada Tabel 6.13 berikut.

Tabel 6.13 Kasus Uji Melihat *List* Pendaftaran Berobat

Nama Kasus Uji	Kasus Uji Melihat <i>List</i> Pendaftaran Berobat
Kode Kebutuhan	RME_F_004
Prosedur	Aktor melakukan <i>log in</i> sebagai petugas pendaftaran
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel data pasien yang telah mendaftar untuk berobat
Hasil	Sistem menampilkan halaman yang berisi tabel data pasien yang telah mendaftar untuk berobat
Status	Valid

Kasus uji melihat *list* pendaftaran berobat alternatif 1 ditunjukkan pada Tabel 6.14 berikut.

Tabel 6.14 Kasus Uji Melihat *List* Pendaftaran Berobat Alternatif 1 : Data pendaftaran berobat kosong

Nama Kasus Uji	Kasus Uji Melihat <i>List</i> Pendaftaran Berobat
Kode Kebutuhan	RME_F_004
Prosedur	Aktor melakukan <i>log in</i> sebagai petugas pendaftaran
Hasil yang diharapkan	Sistem menampilkan tulisan " <i>No data available in table</i> " pada badan tabel pendaftaran berobat
Hasil	Sistem menampilkan tulisan " <i>No data available in table</i> " pada badan tabel pendaftaran berobat
Status	Valid

Kasus uji melihat jumlah kunjungan rawat jalan ditunjukkan pada Tabel 6.15 berikut.

Tabel 6.15 Kasus Uji Melihat Jumlah Kunjungan Rawat Jalan

Nama Kasus Uji	Kasus Uji Melihat Jumlah Kunjungan Rawat Jalan
Kode Kebutuhan	RME_F_005
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas pendaftaran 2. Aktor menekan tombol “Jumlah Kunjungan” pada halaman utama petugas pendaftaran 3. Aktor memilih poli, bulan, dan tahun
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel jumlah kunjungan pasien setiap poli pada bulan yang telah dipilih.
Hasil	Sistem menampilkan halaman yang berisi tabel jumlah kunjungan pasien setiap poli pada bulan yang telah dipilih
Status	Valid

Kasus uji melihat jumlah kunjungan rawat jalan alternatif 1 ditunjukkan pada Tabel 6.16 berikut.

**Tabel 6.16 Kasus Uji Melihat Jumlah Kunjungan Rawat Jalan Alternatif 1 :
Data jumlah kunjungan rawat jalan kosong**

Nama Kasus Uji	Kasus Uji Melihat Jumlah Kunjungan Rawat Jalan
Kode Kebutuhan	RME_F_005
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas pendaftaran 2. Aktor menekan tombol “Jumlah Kunjungan” pada halaman utama petugas pendaftaran 3. Aktor memilih poli, bulan, dan tahun
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data tidak ditemukan
Hasil	Sistem menampilkan <i>alert</i> bahwa data tidak ditemukan
Status	Valid

Kasus uji cetak jumlah kunjungan rawat jalan ditunjukkan pada Tabel 6.17 berikut.

Tabel 6.17 Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan

Nama Kasus Uji	Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan
Kode Kebutuhan	RME_F_006
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas pendaftaran 2. Aktor menekan tombol “Jumlah Kunjungan” pada halaman utama petugas pendaftaran 3. Aktor memilih poli, bulan, dan tahun

Tabel 6.17 Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan

	<ol style="list-style-type: none"> Aktor menekan tombol “cetak” Aktor menekan tombol “print” pada halaman <i>lay out</i> untuk mencetak
Hasil yang diharapkan	Sistem berhasil mencetak jumlah kunjungan pasien kemudian kembali ke halaman yang menampilkan diagram jumlah kunjungan rawat jalan
Hasil	Sistem berhasil mencetak jumlah kunjungan pasien kemudian kembali ke halaman yang menampilkan diagram jumlah kunjungan rawat jalan
Status	Valid

Kasus uji cetak jumlah kunjungan rawat jalan alternatif 1 ditunjukkan pada Tabel 6.18 berikut.

Tabel 6.18 Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan Alternatif 1 : Aktor menekan tombol “Batal” pada halaman layout untuk mencetak

Nama Kasus Uji	Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan
Kode Kebutuhan	RME_F_006
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai petugas pendaftaran Aktor menekan tombol “Jumlah Kunjungan” pada halaman utama petugas pendaftaran Aktor memilih poli, bulan, dan tahun Aktor menekan tombol “cetak” Aktor menekan tombol “Batal” pada halaman <i>lay out</i> untuk mencetak
Hasil yang diharapkan	Sistem kembali menampilkan halaman diagram jumlah kunjungan rawat jalan
Hasil	Sistem kembali menampilkan halaman diagram jumlah kunjungan rawat jalan
Status	Valid

Kasus uji melihat peringkat diagnosa ditunjukkan pada Tabel 6.19 berikut.

Tabel 6.19 Kasus Uji Melihat Peringkat Diagnosa

Nama Kasus Uji	Kasus Uji Melihat Peringkat Diagnosa
Kode Kebutuhan	RME_F_007
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai petugas pendaftaran Aktor menekan tombol “Peringkat Diagnosa” pada halaman utama petugas pendaftaran Aktor memilih poli, bulan, dan tahun
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel peringkat diagnosa pada poli, bulan, dan tahun yang telah dipilih

Tabel 6.19 Kasus Uji Melihat Peringkat Diagnosa

Hasil	Sistem menampilkan halaman yang berisi tabel peringkat diagnosa pada poli, bulan, dan tahun yang telah dipilih
Status	Valid

Kasus uji cetak peringkat diagnosa ditunjukkan pada Tabel 6.20 berikut.

Tabel 6.20 Kasus Uji Cetak Peringkat Diagnosa

Nama Kasus Uji	Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan
Kode Kebutuhan	RME_F_008
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas pendaftaran 2. Aktor menekan tombol “Peringkat Diagnosa” pada halaman utama petugas pendaftaran 3. Aktor memilih poli, bulan, dan tahun 4. Aktor menekan tombol “cetak” 5. Aktor menekan tombol “cetak” pada halaman <i>lay out</i> untuk mencetak
Hasil yang diharapkan	Sistem berhasil mencetak jumlah kunjungan pasien kemudian kembali ke halaman yang menampilkan diagram peringkat diagnosa
Hasil	Sistem berhasil mencetak jumlah kunjungan pasien kemudian kembali ke halaman yang menampilkan diagram peringkat diagnosa
Status	Valid

Kasus uji cetak peringkat diagnosa alternatif 1 ditunjukkan pada Tabel 6.21 berikut.

Tabel 6.21 Kasus Uji Cetak Peringkat Diagnosa Alternatif 1 : Aktor menekan tombol “Batal” pada halaman layout untuk mencetak

Nama Kasus Uji	Kasus Uji Cetak Jumlah Kunjungan Rawat Jalan
Kode Kebutuhan	RME_F_008
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas pendaftaran 2. Aktor menekan tombol “Peringkat Diagnosa” pada halaman utama petugas pendaftaran 3. Aktor memilih poli, bulan, dan tahun 4. Aktor menekan tombol “cetak” 5. Aktor menekan tombol “Batal” pada halaman <i>lay out</i> untuk mencetak
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi diagram peringkat diagnosa

Tabel 6.21 Kasus Uji Cetak Peringkat Diagnosa Alternatif 1 : Aktor menekan tombol “Batal” pada halaman layout untuk mencetak

Hasil	Sistem menampilkan halaman yang berisi diagram peringkat diagnosa
Status	Valid

Kasus uji mengubah status konfirmasi pendaftaran ditunjukkan pada Tabel 6.22 berikut.

Tabel 6.22 Kasus Uji Mengubah Status Konfirmasi Pendaftaran

Nama Kasus Uji	Kasus Uji Mengubah Status Konfirmasi Pendaftaran
Kode Kebutuhan	RME_F_009
Prosedur	1. Aktor melakukan <i>log in</i> sebagai petugas pendaftaran 2. Aktor menekan tombol “Konfirmasi” pada tabel pendaftaran berobat pasien
Hasil yang diharapkan	Status konfirmasi pasien telah berubah dari yang awalnya “Belum Konfirmasi” menjadi “Sudah Konfirmasi”
Hasil	Status konfirmasi pendaftaran telah berubah dari yang awalnya “Belum Konfirmasi” menjadi “Sudah Konfirmasi”
Status	Valid

Kasus uji membuat rekam medis baru ditunjukkan pada Tabel 6.23 berikut.

Tabel 6.23 Kasus Uji Membuat Rekam Medis Baru

Nama Kasus Uji	Kasus Uji Membuat Rekam Medis Baru
Kode Kebutuhan	RME_F_010
Prosedur	1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor menekan tombol “Buat Rekam Medis Baru” 3. Aktor mengisi <i>form</i> rekam medis baru secara lengkap 4. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa pembuatan rekam medis berhasil
Hasil	Sistem menampilkan <i>alert</i> bahwa pembuatan rekam medis berhasil
Status	Valid

Kasus uji membuat rekam medis baru alternatif 1 ditunjukkan pada Tabel 6.24 berikut.

Tabel 6.24 Kasus Uji Membuat Rekam Medis Baru Alternatif 1 : Validasi *form* gagal

Nama Kasus Uji	Kasus Uji Membuat Rekam Medis Baru
-----------------------	------------------------------------

Tabel 6.24 Kasus Uji Membuat Rekam Medis Baru Alternatif 1 : Validasi *form* gagal

Kode Kebutuhan	RME_F_010
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor menekan tombol “Buat Rekam Medis Baru” 3. Aktor mengisi <i>form</i> rekam medis baru dengan mengosongkan satu <i>field</i> 4. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem kembali menampilkan <i>form</i> rekam medis baru beserta pesan kesalahan “ <i>Field ... tidak boleh kosong</i> ” pada <i>field</i> yang belum diisi
Hasil	Sistem kembali menampilkan <i>form</i> rekam medis baru beserta pesan kesalahan “ <i>Field ... tidak boleh kosong</i> ” pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji membuat rekam medis baru alternatif 2 ditunjukkan pada Tabel 6.25 berikut.

Tabel 6.25 Kasus Uji Membuat Rekam Medis Baru Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Membuat Rekam Medis Baru
Kode Kebutuhan	RME_F_010
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor menekan tombol “Buat Rekam Medis Baru” 3. Aktor mengisi <i>form</i> rekam medis baru secara lengkap 4. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat data pribadi pasien ditunjukkan pada Tabel 6.26 berikut.

Tabel 6.26 Kasus Uji Melihat Data Pribadi Pasien

Nama Kasus Uji	Kasus Uji Melihat Data Pribadi Pasien
Kode Kebutuhan	RME_F_011
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis atau sebagai perawat 2. Aktor menekan tombol “Cari Rekam Medis” 3. Aktor mengetikkan nama pasien pada kolom pencarian 4. Aktor menekan tombol “Cari” 5. Aktor menekan tombol “Buka” pada tabel hasil pencarian rekam medis

Tabel 6.26 Kasus Uji Melihat Data Pribadi Pasien

Hasil yang diharapkan	Sistem menampilkan data pribadi yang ada di dalam rekam medis pasien
Hasil	Sistem menampilkan data pribadi yang ada di dalam rekam medis pasien
Status	Valid

Kasus uji melihat data pribadi pasien alternatif 1 ditunjukkan pada Tabel 6.27 berikut.

Tabel 6.27 Kasus Uji Melihat Data Pribadi Pasien Alternatif 1 : Data pencarian rekam medis tidak ditemukan

Nama Kasus Uji	Kasus Uji Melihat Data Pribadi Pasien
Kode Kebutuhan	RME_F_011
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis atau sebagai perawat 2. Aktor menekan tombol “Cari Rekam Medis” 3. Aktor mengetikkan nama pasien pada kolom pencarian 4. Aktor menekan tombol “Cari” 5. Aktor menekan tombol “Buka” pada tabel hasil pencarian rekam medis
Hasil yang diharapkan	Sistem menampilkan pesan bahwa rekam medis tidak ditemukan
Hasil	Sistem menampilkan pesan bahwa rekam medis tidak ditemukan
Status	Valid

Kasus uji mengubah data pribadi pasien ditunjukkan pada Tabel 6.28 berikut.

Tabel 6.28 Kasus Uji Mengubah Data Pribadi Pasien

Nama Kasus Uji	Kasus Uji Mengubah Data Pribadi Pasien
Kode Kebutuhan	RME_F_012
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor membuka <i>file</i> rekam medis pasien 3. Aktor menekan tombol “Edit” pada halaman yang menampilkan data pribadi pasien 4. Aktor mengisi <i>form</i> perubahan data pribadi pasien secara lengkap 5. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa pengubahan data pribadi pasien berhasil dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa pengubahan data pribadi pasien berhasil dan kembali pada halaman rekam medis

Tabel 6.28 Kasus Uji Mengubah Data Pribadi Pasien

Status	Valid
---------------	-------

Kasus uji mengubah data pribadi pasien alternatif 1 ditunjukkan pada Tabel 6.29 berikut.

Tabel 6.29 Kasus Uji Mengubah Data Pribadi Pasien Alternatif 1 : Validasi form gagal

Nama Kasus Uji	Kasus Uji Mengubah Data Pribadi Pasien
Kode Kebutuhan	RME_F_012
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor membuka <i>file</i> rekam medis pasien 3. Aktor menekan tombol “Edit” pada halaman yang menampilkan data pribadi pasien 4. Aktor mengisi <i>form</i> perubahan data pribadi pasien dengan mengosongkan satu <i>field</i> 5. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem kembali menampilkan <i>form</i> rekam medis baru beserta pesan kesalahan “Field ... tidak boleh kosong” pada <i>field</i> yang belum diisi
Hasil	Sistem kembali menampilkan <i>form</i> rekam medis baru beserta pesan kesalahan “Field ... tidak boleh kosong” pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji mengubah data pribadi pasien alternatif 2 ditunjukkan pada Tabel 6.30 berikut.

Tabel 6.30 Kasus Uji Mengubah Data Pribadi Pasien Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Mengubah Data Pribadi Pasien
Kode Kebutuhan	RME_F_012
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor membuka <i>file</i> rekam medis pasien 3. Aktor menekan tombol “Edit” pada halaman yang menampilkan data pribadi pasien 4. Aktor mengisi <i>form</i> perubahan data pribadi pasien secara lengkap 5. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji menghapus rekam medis ditunjukkan pada Tabel 6.31 berikut.

Tabel 6.31 Kasus Uji Menghapus Rekam Medis

Nama Kasus Uji	Kasus Uji Menghapus Rekam Medis
Kode Kebutuhan	RME_F_013
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor mengetikkan nama pasien pada kolom pencarian 3. Aktor memilih tombol “Hapus” pada data yang dipilih 4. Aktor menekan tombol “Oke” pada kotak dialog menghapus data
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa penghapusan rekam medis berhasil dan kembali pada halaman pencarian rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa penghapusan rekam medis berhasil dan kembali pada halaman pencarian rekam medis
Status	Valid

Kasus uji menghapus rekam medis alternatif 1 ditunjukkan pada Tabel 6.32 berikut.

Tabel 6.32 Kasus Uji Menghapus Rekam Medis Alternatif 1 : Aktor menekan tombol “Batal” pada kotak dialog menghapus data

Nama Kasus Uji	Kasus Uji Menghapus Rekam Medis Pasien
Kode Kebutuhan	RME_F_013
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor mengetikkan nama pasien pada kolom pencarian 3. Aktor memilih tombol “Hapus” pada data yang dipilih 4. Aktor menekan tombol “Batal” pada kotak dialog menghapus data
Hasil yang diharapkan	Sistem kembali menampilkan halaman pencarian rekam medis
Hasil	Sistem kembali menampilkan halaman pencarian rekam medis
Status	Valid

Kasus uji menghapus rekam medis alternatif 2 ditunjukkan pada Tabel 6.33 berikut.

Tabel 6.33 Kasus Uji Menghapus Rekam Medis Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Hapus Pendaftaran Berobat
Kode Kebutuhan	RME_F_013
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas rekam medis 2. Aktor mengetikkan nama pasien pada kolom pencarian 3. Aktor memilih tombol “Hapus” pada data yang dipilih

Tabel 6.33 Kasus Uji Menghapus Rekam Medis Alternatif 2 : Koneksi dengan basis data terputus

	1. Aktor menekan tombol “Oke” pada kotak dialog menghapus data
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat *list* pendaftaran pasien per poli ditunjukkan pada Tabel 6.34 berikut.

Tabel 6.34 Kasus Uji Melihat *List* Pendaftaran Pasien Per Poli

Nama Kasus Uji	Kasus Uji Melihat <i>List</i> Pendaftaran Pasien Per Poli
Kode Kebutuhan	RME_F_014
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor memilih poli yang tersedia yaitu poli umum, poli gigi, dan kesehatan ibu dan anak
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi data pasien yang telah mendaftar untuk berobat pada poli yang telah dipilih
Hasil	Sistem menampilkan halaman yang berisi data pasien yang telah mendaftar untuk berobat pada poli yang telah dipilih
Status	Valid

Kasus uji mengubah status antrian pasien ditunjukkan pada Tabel 6.35 berikut.

Tabel 6.35 Kasus Uji Mengubah Status Antrian Pasien

Nama Kasus Uji	Kasus Uji Mengubah Status Antrian Pasien
Kode Kebutuhan	RME_F_015
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor memilih poli yang tersedia yaitu poli umum, poli gigi, dan kesehatan ibu dan anak 3. Aktor menekan tombol “Ubah” pada tabel antrian pelayanan pasien
Hasil yang diharapkan	Status antrian pasien telah berubah dari yang awalnya “Belum Dilayani” menjadi “Sudah Dilayani”
Hasil	Status antrian pasien telah berubah dari yang awalnya “Belum Dilayani” menjadi “Sudah Dilayani”
Status	Valid

Kasus uji melihat rekam medis pasien ditunjukkan pada Tabel 6.36 berikut.

Tabel 6.36 Kasus Uji Melihat Rekam Medis Pasien

Nama Kasus Uji	Kasus Uji Melihat Rekam Medis Pasien
Kode Kebutuhan	RME_F_016
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor memilih poli yang tersedia yaitu poli umum, poli gigi, dan kesehatan ibu dan anak 3. Aktor menekan tombol “Cari rekam medis” pada halaman pendaftaran berobat pasien 4. Aktor mengetikkan nama pasien pada kolom pencarian 5. Aktor menekan tombol “Buka” pada halaman hasil pencarian rekam medis pasien
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi rekam medis pasien yang terdiri dari data pribadi pasien, pengkajian klinis medis, kesehatan umum, kesehatan gigi dan mulut, edukasi rawat jalan, <i>tab</i> riwayat medis
Hasil	Sistem menampilkan halaman yang berisi rekam medis pasien yang terdiri dari data pribadi pasien, pengkajian klinis medis, kesehatan umum, kesehatan gigi dan mulut, edukasi rawat jalan, <i>tab</i> riwayat medis
Status	Valid

Kasus uji melihat rekam medis pasien alternatif 1 ditunjukkan pada Tabel 6.37 berikut.

Tabel 6.37 Kasus Uji Melihat Rekam Medis Pasien Alternatif 1 : Data pencarian rekam medis tidak ditemukan

Nama Kasus Uji	Kasus Uji Melihat Rekam Medis Pasien
Kode Kebutuhan	RME_F_016
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor memilih poli yang tersedia yaitu poli umum, poli gigi, dan kesehatan ibu dan anak 3. Aktor menekan tombol “Cari rekam medis” pada halaman pendaftaran berobat pasien 4. Aktor mengetikkan nama pasien pada kolom pencarian
Hasil yang diharapkan	Sistem menampilkan pesan bahwa rekam medis tidak ditemukan
Hasil	Sistem menampilkan pesan bahwa rekam medis tidak ditemukan
Status	Valid

Kasus uji melihat riwayat pengkajian klinis medis ditunjukkan pada Tabel 6.38 berikut.

Tabel 6.38 Kasus Uji Melihat Riwayat Pengkajian Klinis Medis

Nama Kasus Uji	Kasus Uji Melihat Riwayat Pengkajian Klinis Medis
Kode Kebutuhan	RME_F_017
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Pengkajian Klinis Medis pada rekam medis pasien
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel riwayat pengkajian klinis medis rawat jalan pasien
Hasil	Sistem menampilkan halaman yang berisi tabel riwayat pengkajian klinis medis rawat jalan pasien
Status	Valid

Kasus uji melihat riwayat pengkajian klinis medis alternatif 1 ditunjukkan pada Tabel 6.39 berikut.

**Tabel 6.39 Kasus Uji Melihat Riwayat Pengkajian Klinis Medis Alternatif 1 :
Data pengkajian klinis medis kosong**

Nama Kasus Uji	Kasus Uji Melihat Riwayat Pengkajian Klinis Medis
Kode Kebutuhan	RME_F_017
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Pengkajian Klinis Medis pada rekam medis pasien
Hasil yang diharapkan	Sistem menampilkan tabel riwayat pengkajian klinis medis yang masih kosong
Hasil	Sistem menampilkan tabel riwayat pengkajian klinis medis yang masih kosong
Status	Valid

Kasus uji melihat detail data pengkajian klinis medis ditunjukkan pada Tabel 6.40 berikut.

Tabel 6.40 Kasus Uji Melihat Detail Data Pengkajian Klinis Medis

Nama Kasus Uji	Kasus Uji Melihat Detail Data Pengkajian Klinis Medis
Kode Kebutuhan	RME_F_018
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Pengkajian Klinis Medis pada rekam medis pasien 4. Aktor memilih data pengkajian klinis medis yang ingin dibuka pada tabel daftar pengkajian klinis medis rawat jalan

Tabel 6.40 Kasus Uji Melihat Detail Data Pengkajian Klinis Medis

	5. Aktor menekan tombol “Buka” pada data yang dipilih
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi detail pengkajian klinis medis rawat jalan pasien
Hasil	Sistem menampilkan halaman yang berisi detail pengkajian klinis medis rawat jalan pasien
Status	Valid

Kasus uji membuat pengkajian klinis medis rawat jalan baru ditunjukkan pada Tabel 6.41 berikut.

Tabel 6.41 Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru

Nama Kasus Uji	Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru
Kode Kebutuhan	RME_F_019
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih tab Pengkajian Klinis Medis pada rekam medis pasien 4. Aktor memilih menu “Tambah Data” 5. Aktor mengisi <i>form</i> pengkajian klinis medis rawat jalan 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji membuat pengkajian klinis medis rawat jalan baru alternatif 1 ditunjukkan pada Tabel 6.42 berikut.

Tabel 6.42 Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru Alternatif 1 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru
Kode Kebutuhan	RME_F_019
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih tab Pengkajian Klinis Medis pada rekam medis pasien 4. Aktor memilih menu “Tambah Data” 5. Aktor mengisi <i>form</i> pengkajian klinis medis rawat jalan 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data

Tabel 6.42 Kasus Uji Membuat Pengkajian Klinis Medis Rawat Jalan Baru Alternatif 1 : Koneksi dengan basis data terputus

Status	Valid
--------	-------

Kasus uji membuat data rawat jalan umum baru ditunjukkan pada Tabel 6.43 berikut.

Tabel 6.43 Kasus Uji Membuat Data Rawat Jalan Umum Baru

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Umum Baru
Kode Kebutuhan	RME_F_020
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum 4. Aktor menekan tombol "Buat Data Baru" 5. Aktor mengisi <i>form</i> rawat jalan umum secara lengkap 6. Aktor menekan tombol "Simpan"
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji membuat data rawat jalan umum baru alternatif 1 ditunjukkan pada Tabel 6.44 berikut.

Tabel 6.44 Kasus Uji Membuat Data Rawat Jalan Umum Baru Alternatif 1: Ada *field* pada *form* yang belum terisi

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Umum Baru
Kode Kebutuhan	RME_F_020
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum 4. Aktor menekan tombol "Buat Data Baru" 5. Aktor mengisi <i>form</i> rawat jalan umum baru dengan mengosongkan salah satu <i>field</i> 6. Aktor menekan tombol "Simpan"
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji membuat data rawat jalan umum baru alternatif 2 ditunjukkan pada Tabel 6.45 berikut.

**Tabel 6.45 Kasus Uji Membuat Data Rawat Jalan Umum Baru Alternatif 2:
Koneksi dengan basis data terputus**

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Umum Baru
Kode Kebutuhan	RME_F_020
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum 4. Aktor menekan tombol “Buat Data Baru” 5. Aktor mengisi <i>form</i> rawat jalan umum secara lengkap 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji membuat data rawat jalan gigi baru ditunjukkan pada Tabel 6.46 berikut.

Tabel 6.46 Kasus Uji Membuat Data Rawat Jalan Gigi Baru

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Gigi Baru
Kode Kebutuhan	RME_F_021
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut 4. Aktor menekan tombol “Buat Data Baru” 5. Aktor mengisi mengisi <i>form</i> rawat jalan gigi secara lengkap 6. Aktor mengunggah <i>file</i> foto odontogram 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji membuat data rawat jalan gigi baru alternatif 1 ditunjukkan pada Tabel 6.47 berikut.

**Tabel 6.47 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 1: Ada
field pada form yang belum terisi**

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Gigi Baru
Kode Kebutuhan	RME_F_021
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat

Tabel 6.47 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 1: Ada *field* pada *form* yang belum terisi

	<ol style="list-style-type: none"> Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut Aktor menekan tombol “Buat Data Baru” Aktor mengisi mengisi <i>form</i> rawat jalan gigi dengan mengosongkan salah satu <i>field</i> Aktor mengunggah <i>file</i> foto odontogram Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji membuat data rawat jalan gigi baru alternatif 2 ditunjukkan pada Tabel 6.48 berikut.

Tabel 6.48 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 2: Format *file* foto salah atau ukuran terlalu besar

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Gigi Baru
Kode Kebutuhan	RME_F_021
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut Aktor menekan tombol “Buat Data Baru” Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap Aktor mengunggah <i>file</i> foto odontogram Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan format <i>file</i> foto atau kesalahan ukuran
Hasil	Sistem menampilkan pesan kesalahan format <i>file</i> foto atau kesalahan ukuran
Status	Valid

Kasus uji membuat data rawat jalan gigi baru alternatif 3 ditunjukkan pada Tabel 6.49 berikut.

Tabel 6.49 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 3: Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Membuat Data Rawat Jalan Gigi Baru
Kode Kebutuhan	RME_F_021
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat

**Tabel 6.49 Kasus Uji Membuat Data Rawat Jalan Gigi Baru Alternatif 3:
Koneksi dengan basis data terputus**

	<ol style="list-style-type: none"> Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut Aktor menekan tombol “Buat Data Baru” Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat detail data rawat jalan umum ditunjukkan pada Tabel 6.50 berikut.

Tabel 6.50 Kasus Uji Melihat Detail Data Rawat Jalan Umum

Nama Kasus Uji	Kasus Uji Melihat Detail Data Rawat Jalan Umum
Kode Kebutuhan	RME_F_022
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Umum Aktor memilih menu “Buka” pada data kunjungan yang dipilih dari tabel riwayat kunjungan
Hasil yang diharapkan	Sistem menampilkan data rawat jalan pasien yang telah dipilih
Hasil	Sistem menampilkan data rawat jalan pasien yang telah dipilih
Status	Valid

Kasus uji melihat detail data rawat jalan gigi ditunjukkan pada Tabel 6.51 berikut.

Tabel 6.51 Kasus Uji Melihat Detail Data Rawat Jalan Gigi

Nama Kasus Uji	Kasus Uji Melihat Detail Data Rawat Jalan Gigi
Kode Kebutuhan	RME_F_023
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut Aktor memilih menu “Buka” pada data kunjungan yang dipilih dari tabel riwayat kunjungan
Hasil yang diharapkan	Sistem menampilkan data rawat jalan pasien yang telah dipilih
Hasil	Sistem menampilkan data rawat jalan pasien yang telah dipilih
Status	Valid

Kasus uji menambah data rawat jalan umum ditunjukkan pada Tabel 6.52 berikut.

Tabel 6.52 Kasus Uji Menambah Data Rawat Jalan Umum

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Umum
Kode Kebutuhan	RME_F_024
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi semua <i>form</i> rawat jalan umum secara lengkap 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji menambah data rawat jalan umum alternatif 1 ditunjukkan pada Tabel 6.53 berikut.

Tabel 6.53 Kasus Uji Menambah Data Rawat Jalan Umum Alternatif 1 : Ada *field* pada *form* yang belum terisi

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Umum
Kode Kebutuhan	RME_F_024
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi <i>form</i> rawat jalan umum dengan mengosongkan salah satu <i>field</i> 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji menambah data rawat jalan umum alternatif 2 ditunjukkan pada Tabel 6.54 berikut.

Tabel 6.54 Kasus Uji Menambah Data Rawat Jalan Umum Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Umum
Kode Kebutuhan	RME_F_024

Tabel 6.54 Kasus Uji Menambah Data Rawat Jalan Umum Alternatif 2 : Koneksi dengan basis data terputus

Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi <i>form</i> rawat jalan umum secara lengkap 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji menambah data rawat jalan gigi ditunjukkan pada Tabel 6.55 berikut.

Tabel 6.55 Kasus Uji Menambah Data Rawat Jalan Gigi

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Gigi
Kode Kebutuhan	RME_F_025
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap 6. Aktor mengunggah <i>file</i> foto odontogram 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji menambah data rawat jalan gigi alternatif 1 ditunjukkan pada Tabel 6.56 berikut.

Tabel 6.56 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 1 : Ada *field* pada *form* yang belum terisi

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Gigi
Kode Kebutuhan	RME_F_025
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi <i>form</i> rawat jalan gigi dengan mengosongkan salah satu <i>field</i>

Tabel 6.56 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 1 : Ada *field* pada *form* yang belum terisi

	6. Aktor mengunggah <i>file</i> foto odontogram 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji menambah data rawat jalan gigi alternatif 2 ditunjukkan pada Tabel 6.57 berikut.

Tabel 6.57 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 2 Format *file* foto salah atau ukuran terlalu besar

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Gigi
Kode Kebutuhan	RME_F_025
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap 6. Aktor mengunggah <i>file</i> foto odontogram 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan format <i>file</i> foto atau kesalahan ukuran
Hasil	Sistem menampilkan pesan kesalahan format <i>file</i> foto atau kesalahan ukuran
Status	Valid

Kasus uji menambah data rawat jalan gigi alternatif 3 ditunjukkan pada Tabel 6.58 berikut.

Tabel 6.58 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 3: Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Menambah Data Rawat Jalan Gigi
Kode Kebutuhan	RME_F_025
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut 4. Aktor menekan tombol “Tambah” 5. Aktor mengisi <i>form</i> rawat jalan gigi secara lengkap 6. Aktor menekan tombol “Simpan”

Tabel 6.58 Kasus Uji Menambah Data Rawat Jalan Gigi Alternatif 3: Koneksi dengan basis data terputus

Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat riwayat kunjungan umum ditunjukkan pada Tabel 6.59 berikut.

Tabel 6.59 Kasus Uji Melihat Riwayat Kunjungan Umum

Nama Kasus Uji	Kasus Uji Melihat Riwayat Kunjungan Umum
Kode Kebutuhan	RME_F_026
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi data riwayat kunjungan pasien dalam bentuk tabel
Hasil	Sistem menampilkan halaman yang berisi data riwayat kunjungan pasien dalam bentuk tabel
Status	Valid

Kasus uji melihat riwayat kunjungan umum alternatif 1 ditunjukkan pada Tabel 6.60 berikut.

Tabel 6.60 Kasus Uji Melihat Riwayat Kunjungan Umum Alternatif 1 : Data riwayat kunjungan masih kosong

Nama Kasus Uji	Kasus Uji Melihat Riwayat Kunjungan Umum
Kode Kebutuhan	RME_F_026
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Kesehatan Umum
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel riwayat kunjungan yang masih kosong
Hasil	Sistem menampilkan halaman yang berisi tabel riwayat kunjungan yang masih kosong
Status	Valid

Kasus uji melihat riwayat kunjungan gigi ditunjukkan pada Tabel 6.61 berikut.

Tabel 6.61 Kasus Uji Melihat Riwayat Kunjungan Gigi

Nama Kasus Uji	Kasus Uji Melihat Riwayat Kunjungan Gigi
Kode Kebutuhan	RME_F_027
Prosedur	1. Aktor melakukan <i>log in</i> sebagai perawat

Tabel 6.61 Kasus Uji Melihat Riwayat Kunjungan Gigi

	<ol style="list-style-type: none"> Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Gigi dan Mulut
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi data riwayat kunjungan pasien dalam bentuk tabel
Hasil	Sistem menampilkan halaman yang berisi data riwayat kunjungan pasien dalam bentuk tabel
Status	Valid

Kasus uji melihat riwayat kunjungan gigi alternatif 1 ditunjukkan pada Tabel 6.62 berikut.

Tabel 6.62 Kasus Uji Melihat Riwayat Kunjungan Gigi Alternatif 1: Data riwayat kunjungan masih kosong

Nama Kasus Uji	Kasus Uji Melihat Riwayat Kunjungan Gigi
Kode Kebutuhan	RME_F_027
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Kesehatan Umum
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel riwayat kunjungan yang masih kosong
Hasil	Sistem menampilkan halaman yang berisi tabel riwayat kunjungan yang masih kosong
Status	Valid

Kasus uji melihat riwayat farmasi ditunjukkan pada Tabel 6.63 berikut.

Tabel 6.63 Kasus Uji Melihat Riwayat Farmasi

Nama Kasus Uji	Kasus Uji Melihat Riwayat Farmasi
Kode Kebutuhan	RME_F_028
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Riwayat Medis Aktor memilih menu “Riwayat Obat” pada <i>tab</i> Riwayat Medis
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi data riwayat terapi farmasi pasien dalam bentuk tabel
Hasil	Sistem menampilkan halaman yang berisi data riwayat terapi farmasi pasien dalam bentuk tabel
Status	Valid

Kasus uji melihat riwayat farmasi alternatif 1 ditunjukkan pada Tabel 6.64 berikut.

Tabel 6.64 Kasus Uji Melihat Riwayat Farmasi Alternatif 1 : Data riwayat farmasi masih kosong

Nama Kasus Uji	Kasus Uji Melihat Riwayat Farmasi
Kode Kebutuhan	RME_F_028
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Obat” pada <i>tab</i> Riwayat Medis
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel riwayat farmasi yang masih kosong
Hasil	Sistem menampilkan halaman yang berisi tabel riwayat farmasi yang masih kosong
Status	Valid

Kasus uji menambah data terapi farmasi ditunjukkan pada Tabel 6.65 berikut.

Tabel 6.65 Kasus Uji Menambah Data Terapi Farmasi

Nama Kasus Uji	Kasus Uji Menambah Data Terapi Farmasi
Kode Kebutuhan	RME_F_029
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Obat” 5. Aktor menekan tombol “Tambah Data” 6. Aktor mengisi <i>form</i> terapi farmasi seccara lengkap 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji menambah data terapi farmasi alternatif 1 ditunjukkan pada Tabel 6.66 berikut.

Tabel 6.66 Kasus Uji Menambah Data Terapi Farmasi Alternatif 1: Validasi *form* gagal

Nama Kasus Uji	Kasus Uji Menambah Data Terapi Farmasi
Kode Kebutuhan	RME_F_029
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis

Tabel 6.66 Kasus Uji Menambah Data Terapi Farmasi Alternatif 1: Validasi form gagal

	<ol style="list-style-type: none"> Aktor memilih menu “Riwayat Obat” Aktor menekan tombol “Tambah Data” Aktor mengisi <i>form</i> terapi farmasi dengan mengosongkan salah satu <i>field</i> Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji menambah data terapi farmasi alternatif 2 ditunjukkan pada Tabel 6.67 berikut.

Tabel 6.67 Kasus Uji Menambah Data Terapi Farmasi Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Menambah Data Terapi Farmasi
Kode Kebutuhan	RME_F_029
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Riwayat Medis Aktor memilih menu “Riwayat Obat” Aktor menekan tombol “Tambah Data” Aktor mengisi <i>form</i> terapi farmasi secara lengkap Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat detail terapi farmasi ditunjukkan pada Tabel 6.68 berikut.

Tabel 6.68 Kasus Uji Melihat Detail Terapi Farmasi

Nama Kasus Uji	Kasus Uji Melihat Detail Terapi Farmasi
Kode Kebutuhan	RME_F_030
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai perawat Aktor melihat rekam medis pasien Aktor memilih <i>tab</i> Riwayat Medis Aktor memilih menu “Riwayat Obat” Aktor memilih data yang akan dilihat detailnya pada tabel riwayat terapi obat Aktor menekan tombol “Detail”

Tabel 6.68 Kasus Uji Melihat Detail Terapi Farmasi

Hasil yang diharapkan	Sistem menampilkan halaman yang berisi detail terapi farmasi yang telah dipilih
Hasil	Sistem menampilkan halaman yang berisi detail terapi farmasi yang telah dipilih
Status	Valid

Kasus uji melihat riwayat diagnostik ditunjukkan pada Tabel 6.69 berikut.

Tabel 6.69 Kasus Uji Melihat Riwayat Diagnostik

Nama Kasus Uji	Kasus Uji Melihat Riwayat Diagnostik
Kode Kebutuhan	RME_F_031
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Diagnostik” pada <i>tab</i> Riwayat Medis
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi data riwayat terapi diagnostik pasien dalam bentuk tabel
Hasil	Sistem menampilkan halaman yang berisi data riwayat terapi diagnostik pasien dalam bentuk tabel
Status	Valid

Kasus uji melihat riwayat diagnostik alternatif 1 ditunjukkan pada Tabel 6.70 berikut.

Tabel 6.70 Kasus Uji Melihat Riwayat Diagnostik Alternatif 1: Data riwayat diagnostik masih kosong

Nama Kasus Uji	Kasus Uji Melihat Riwayat Diagnostik
Kode Kebutuhan	RME_F_031
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Diagnostik” pada <i>tab</i> Riwayat Medis
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel riwayat diagnostik yang masih kosong
Hasil	Sistem menampilkan halaman yang berisi tabel riwayat diagnostik yang masih kosong
Status	Valid

Kasus uji melihat hasil pemeriksaan diagnostik ditunjukkan pada Tabel 6.71 berikut.

Tabel 6.71 Kasus Uji Melihat Hasil Pemeriksaan Diagnostik

Nama Kasus Uji	Kasus Uji Melihat Hasil Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_032
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Diagnostik” pada <i>tab</i> Riwayat Medis 5. Aktor menekan tombol “Lihat Hasil” pada tabel riwayat diagnostik
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi data riwayat terapi diagnostik pasien dalam bentuk tabel
Hasil	Sistem menampilkan halaman yang berisi data riwayat terapi diagnostik pasien dalam bentuk tabel
Status	Valid

Kasus uji menambah permintaan pemeriksaan diagnostik ditunjukkan pada Tabel 6.72 berikut.

Tabel 6.72 Kasus Uji Menambah Permintaan Pemeriksaan Diagnostik

Nama Kasus Uji	Kasus Uji Menambah Permintaan Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_033
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Diagnostik” 5. Aktor menekan tombol “Tambah Data” 6. Aktor mengisi <i>form</i> penambahan data permintaan pemeriksaan diagnostik secara lengkap 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji menambah permintaan pemeriksaan diagnostik alternatif 1 ditunjukkan pada Tabel 6.73 berikut.

**Tabel 6.73 Kasus Uji Menambah Permintaan Pemeriksaan Diagnostik
Alternatif 1 : Koneksi dengan basis data terputus**

Nama Kasus Uji	Kasus Uji Menambah Permintaan Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_033
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Riwayat Medis 4. Aktor memilih menu “Riwayat Diagnostik” 5. Aktor menekan tombol “Tambah Data” 6. Aktor mengisi <i>form</i> permintaan pemeriksaan diagnostik secara lengkap 7. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat riwayat pemberian informasi dan edukasi rawat jalan ditunjukkan pada Tabel 6.74 berikut.

**Tabel 6.74 Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi
Rawat Jalan**

Nama Kasus Uji	Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan
Kode Kebutuhan	RME_F_034
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Edukasi Rawat Jalan
Hasil yang diharapkan	Sistem menampilkan tabel riwayat pemberian informasi dan edukasi rawat jalan
Hasil	Sistem menampilkan tabel riwayat pemberian informasi dan edukasi rawat jalan
Status	Valid

Kasus uji melihat riwayat pemberian informasi dan edukasi rawat jalan alternatif 1 ditunjukkan pada Tabel 6.75 berikut.

**Tabel 6.75 Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi
Rawat Jalan Alternatif 1 : Data riwayat edukasi rawat jalan masih kosong**

Nama Kasus Uji	Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan
Kode Kebutuhan	RME_F_034
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien

Tabel 6.75 Kasus Uji Melihat Riwayat Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 1 : Data riwayat edukasi rawat jalan masih kosong

	3. Aktor memilih <i>tab</i> Edukasi Rawat Jalan
Hasil yang diharapkan	Sistem menampilkan tabel riwayat pemberian edukasi rawat jalan yang masih kosong
Hasil	Sistem menampilkan tabel riwayat pemberian edukasi rawat jalan yang masih kosong
Status	Valid

Kasus uji melihat detail data pemberian informasi dan edukasi rawat jalan ditunjukkan pada Tabel 6.76 berikut.

Tabel 6.76 Kasus Uji Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan

Nama Kasus Uji	Kasus Uji Melihat Detail Data Pemberian Informasi dan Edukasi Rawat Jalan
Kode Kebutuhan	RME_F_035
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Edukasi Rawat Jalan 4. Aktor memilih data yang akan dilihat detailnya pada tabel riwayat edukasi rawat jalan 5. Aktor menekan tombol “Buka”
Hasil yang diharapkan	Sistem menampilkan detail dari data pemberian informasi dan edukasi rawat jalan yang dipilih
Hasil	Sistem menampilkan detail dari data pemberian informasi dan edukasi rawat jalan yang dipilih
Status	Valid

Kasus uji menambah data pemberian informasi dan edukasi rawat jalan ditunjukkan pada Tabel 6.77 berikut.

Tabel 6.77 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan

Nama Kasus Uji	Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan
Kode Kebutuhan	RME_F_036
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Edukasi Rawat Jalan 4. Aktor menekan tombol “Tambah Data” 5. Aktor mengisi <i>form</i> pemberian informasi dan edukasi rawat jalan secara lengkap

Tabel 6.77 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan

	6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Hasil	Sistem menampilkan <i>alert</i> bahwa data berhasil disimpan dan kembali pada halaman rekam medis
Status	Valid

Kasus uji menambah data pemberian informasi dan edukasi rawat jalan alternatif 1 ditunjukkan pada Tabel 6.78 berikut.

Tabel 6.78 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 1 : Validasi *form* gagal

Nama Kasus Uji	Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan
Kode Kebutuhan	RME_F_036
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Edukasi Rawat Jalan 4. Aktor menekan tombol “Tambah Data” 5. Aktor mengisi <i>form</i> pemberian informasi dan edukasi rawat jalan dengan mengosongkan salah satu <i>field</i> 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji menambah data pemberian informasi dan edukasi rawat jalan alternatif 2 ditunjukkan pada Tabel 6.79 berikut.

Tabel 6.79 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 2 : Koneksi dengan basis data terputus

Nama Kasus Uji	Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan
Kode Kebutuhan	RME_F_036
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai perawat 2. Aktor melihat rekam medis pasien 3. Aktor memilih <i>tab</i> Edukasi Rawat Jalan 4. Aktor menekan tombol “Tambah Data”

Tabel 6.79 Kasus Uji Menambah Data Pemberian Informasi dan Edukasi Rawat Jalan Alternatif 2 : Koneksi dengan basis data terputus

	5. Aktor mengisi <i>form</i> pemberian informasi dan edukasi rawat jalan secara lengkap 6. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji melihat daftar permintaan pemeriksaan diagnostik ditunjukkan pada Tabel 6.80 berikut.

Tabel 6.80 Kasus Uji Melihat Daftar Permintaan Pemeriksaan Diagnostik

Nama Kasus Uji	Kasus Uji Melihat Daftar Permintaan Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_037
Prosedur	Aktor melakukan <i>log in</i> sebagai petugas laboratorium
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi tabel daftar permintaan diagnostik
Hasil	Sistem menampilkan halaman yang berisi tabel daftar permintaan diagnostik
Status	Valid

Kasus uji melihat daftar permintaan pemeriksaan diagnostik alternatif 1 ditunjukkan pada Tabel 6.81 berikut.

Tabel 6.81 Kasus Uji Melihat Daftar Permintaan Pemeriksaan Diagnostik Alternatif 1 : Data permintaan pemeriksaan diagnostik kosong

Nama Kasus Uji	Kasus Uji Melihat Daftar Permintaan Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_037
Prosedur	Aktor melakukan <i>log in</i> sebagai petugas laboratorium
Hasil yang diharapkan	Sistem menampilkan data pendaftaran berobat masih kosong, sistem menampilkan tulisan “ <i>No data available in table</i> ” pada badan tabel
Hasil	Sistem menampilkan data pendaftaran berobat masih kosong, sistem menampilkan tulisan “ <i>No data available in table</i> ” pada badan tabel
Status	Valid

Kasus uji melihat detail data permintaan pemeriksaan diagnostik ditunjukkan pada Tabel 6.82 berikut.

Tabel 6.82 Kasus Uji Melihat Detail Data Melihat Detail Permintaan Pemeriksaan Diagnostik

Nama Kasus Uji	Kasus Uji Melihat Detail Permintaan Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_038
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas laboratorium 2. Aktor memilih data yang akan dilihat detailnya pada tabel daftar permintaan diagnostik 3. Aktor menekan tombol “Buka”
Hasil yang diharapkan	Sistem menampilkan halaman yang berisi detail permintaan pemeriksaan diagnostik yang telah dipilih
Hasil	Sistem menampilkan halaman yang berisi detail permintaan pemeriksaan diagnostik yang telah dipilih
Status	Valid

Kasus uji menambah hasil pemeriksaan diagnostik ditunjukkan pada tabel 6.83 berikut.

Tabel 6.83 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik

Nama Kasus Uji	Kasus Uji Menambah Hasil Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_039
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas laboratorium 2. Aktor memilih menu “Input Hasil” pada tabel permintaan pemeriksaan diagnostik 3. Aktor mengisi <i>form</i> hasil pemeriksaan diagnostik secara lengkap 4. Aktor mengunggah <i>file</i> foto hasil pemeriksaan diagnostik 5. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan <i>alert</i> bahwa hasil pemeriksaan diagnostik berhasil disimpan dan kembali pada halaman daftar permintaan pemeriksaan diagnostik
Hasil	Sistem menampilkan <i>alert</i> bahwa hasil pemeriksaan diagnostik berhasil disimpan dan kembali pada halaman daftar permintaan pemeriksaan diagnostik
Status	Valid

Kasus uji menambah hasil pemeriksaan diagnostik alternatif 1 ditunjukkan pada Tabel 6.84 berikut.

Tabel 6.84 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik Alternatif 1 : Validasi *form* gagal

Nama Kasus Uji	Kasus Uji Menambah Hasil Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_039
Prosedur	<ol style="list-style-type: none"> 1. Aktor melakukan <i>log in</i> sebagai petugas laboratorium

**Tabel 6.84 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik Alternatif 1 :
Validasi *form* gagal**

	<ol style="list-style-type: none"> Aktor memilih menu “Input Hasil” pada tabel permintaan pemeriksaan diagnostik Aktor mengisi <i>form</i> hasil pemeriksaan diagnostik dengan mengosongkan salah satu <i>field</i> Aktor mengunggah <i>file</i> foto hasil pemeriksaan diagnostik Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Hasil	Sistem menampilkan pesan kesalahan pada <i>field</i> yang belum diisi
Status	Valid

Kasus uji menambah hasil pemeriksaan diagnostik alternatif 2 ditunjukkan pada Tabel 6.85 berikut.

**Tabel 6.85 Kasus Uji Menambah Hasil Pemeriksaan Diagnostik Alternatif 2 :
Format *file* foto salah atau ukuran *file* terlalu besar**

Nama Kasus Uji	Kasus Uji Menambah Hasil Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_039
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai petugas laboratorium Aktor memilih menu “Input Hasil” pada tabel permintaan pemeriksaan diagnostik Aktor mengisi <i>form</i> hasil pemeriksaan diagnostik secara lengkap Aktor mengunggah <i>file</i> foto hasil pemeriksaan diagnostik Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan format <i>file</i> foto atau kesalahan ukuran
Hasil	Sistem menampilkan pesan kesalahan format <i>file</i> foto atau kesalahan ukuran
Status	Valid

Kasus uji menambah hasil pemeriksaan diagnostik alternatif 3 ditunjukkan pada Tabel 6.86 berikut.

**Tabel 6.86 Kasus Uji Menambah Menambah Hasil Pemeriksaan Diagnostik
Alternatif 3 : Koneksi dengan basis data terputus**

Nama Kasus Uji	Kasus Uji Menambah Hasil Pemeriksaan Diagnostik
Kode Kebutuhan	RME_F_039
Prosedur	<ol style="list-style-type: none"> Aktor melakukan <i>log in</i> sebagai petugas laboratorium

Tabel 6.86 Kasus Uji Menambah Menambah Hasil Pemeriksaan Diagnostik Alternatif 3 : Koneksi dengan basis data terputus

	2. Aktor memilih menu “Input Hasil” pada tabel permintaan pemeriksaan diagnostik 3. Aktor mengisi <i>form</i> hasil pemeriksaan diagnostik secara lengkap 4. Aktor mengunggah <i>file</i> foto hasil pemeriksaan diagnostik 5. Aktor menekan tombol “Simpan”
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan koneksi basis data
Hasil	Sistem menampilkan pesan kesalahan koneksi basis data
Status	Valid

Kasus uji *log out* ditunjukkan pada Tabel 6.87 berikut.

Tabel 6.87 Kasus Uji Log out

Alternatif 2 : Koneksi dengan basis data terputus	
Nama Kasus Uji	Kasus Uji <i>Log out</i>
Kode Kebutuhan	RME_F_003
Prosedur	Aktor memilih tombol “Log out” pada halaman web
Hasil yang diharapkan	Sistem kembali ke halaman awal web
Hasil	Sistem kembali ke halaman awal web
Status	Valid

6.3 Pengujian *Compatibility*

Pengujian *compatibility* dilakukan dengan maksud memeriksa apakah perangkat lunak yang dibuat dapat berjalan pada berbagai jenis *browser* karena sistem digunakan oleh lebih dari satu pengguna sehingga terdapat kemungkinan pengguna mengakses sistem dengan jenis *browser* yang berbeda antara pengguna satu dengan pengguna lainnya. Pada penelitian ini, penulis menggunakan SortSite. Cara pengujian menggunakan SortSite yaitu dengan memasukkan URL web ke SortSite dan memilih jenis pengujian *compatibility*. Setelah menekan tombol “Check”, pengujian dimulai dan didapatkan hasil seperti gambar 6.4 berikut.

Summary Issues Pages <http://local/host/eliasonline/>

Errors Accessibility Compatibility Privacy Search Standards Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser Version	IE 9 10 11	Edge 16	Firefox 60	Safari 11	Chrome 66	iOS 11	Android ≤ 3 4*	BlackBerry 10.0
Critical Issues	✓ ✓ ✓	✓	●	✓	●	✓	✓	✓
Major Issues	✓ ✓ ✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	● ✓ ✓	✓	●	✓	✓	✓	✓	✓

Key

- Missing content or functionality
- Major layout or performance problems
- Minor layout or performance problems

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Priority	Description and URL	Guideline and Line#	Count
Priority 1			
1 issues on 1 pages			
●	Chrome and Firefox display multiple security warnings for password fields that don't use HTTPS.	Chrome ≥ 56 Firefox ≥ 52	1 pages
Priority 3			
3 issues on 2 pages			
●	CSS selectors for INPUT TYPE="SEARCH" are ignored in IE 9 and iOS 4 or earlier.	Internet Explorer ≤ 9	1 pages
●	The page-break-after CSS property values avoid, left and right are not supported by Firefox.	Firefox	1 pages
●	The text-shadow CSS property is not supported by some browsers.	Internet Explorer ≤ 9	1 pages

Gambar 6.4 Hasil Pengujian *Compatibility*

Berdasarkan hasil pengujian *compatibility* pada gambar 6.4, situs web dapat berjalan pada sebelas versi *browser*. Masalah dalam pengujian ini dikelompokkan menjadi tiga kategori yaitu *critical issues*, *major issues*, dan *minor issues*. Terdapat 1 *critical issues* saat situs web dijalankan pada *browser Firefox* versi ≤ 56 dan *Chrome* versi ≤ 52 . Kemudian terdapat *minor issues* sejumlah 3 pada *browser* Internet Explorer versi ≤ 9 dan *Firefox*. *Critical issues* terjadi karena web yang diuji tidak menggunakan HTTPS untuk menjaga keamanan *password* yang dimasukkan pada halaman *log in*. Namun situs web dan fungsi-fungsi didalamnya masih bisa dijalankan walaupun diakses tanpa menggunakan HTTPS.

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil penelitian pembangunan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* berbasis web dengan studi kasus Klinik Medis Elisa Malang, didapatkan kesimpulan sebagai berikut : dari hasil analisis kebutuhan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* di Klinik Elisa Malang, didapatkan 39 kebutuhan fungsional dan 1 kebutuhan non-fungsional. Pada analisis data menggunakan *entity relationship diagram* yang menggambarkan hubungan antara delapan entitas dalam sistem. Pada pemodelan *use case*, diperoleh 39 *use case* yang masing-masing dijelaskan pada *use case scenario*.

Kemudian pada perancangan sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* di Klinik Elisa Malang, diperoleh hasil perancangan *sequence diagram*, perancangan basis data, *class diagram*, perancangan algoritme, dan perancangan antarmuka. *Sequence diagram* menggambarkan urutan interaksi antar objek dalam sebuah fungsi. Perancangan basis data menggunakan *physical data model* yang menggambarkan tabel-tabel dan atribut serta relasi antar tabel. Perancangan *class diagram* menggunakan konsep *Model View Controller* (MVC) yang menghasilkan 9 *class* model dan 9 *class controller*. Pada perancangan antarmuka dihasilkan lima perancangan antarmuka untuk halaman utama tiap pengguna.

Setelah perancangan selesai, dilakukan implementasi sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* di Klinik Elisa Malang yang menghasilkan spesifikasi pengembangan perangkat lunak, implementasi basis data berupa *Data Definition Language* (DDL), implementasi algoritme berupa *source code*, dan implementasi halaman antarmuka. Implementasi kode program dilakukan dengan menggunakan bahasa pemrograman PHP dan JavaScript.

Kemudian dari hasil pengujian sistem aplikasi rekam medis elektronik dan pendaftaran pasien *online* di Klinik Elisa Malang, pengujian unit menggunakan metode *white box* dengan teknik *basis path testing* menghasilkan nilai valid untuk setiap jalur independen dengan presentase keberhasilan 100 %. Kemudian pada pengujian validasi menghasilkan nilai valid pada setiap kasus uji dengan presentase keberhasilan 100%. Pengujian *compatibility* menggunakan SortSite menghasilkan web dapat berjalan dengan baik pada 9 dari 11 versi *browser* dengan presentase keberhasilan 82% .

7.2 Saran

Saran yang diberikan untuk penelitian mengenai sistem rekam medis elektronik dan pendaftaran pasien *online* selanjutnya yaitu pendaftaran pasien *online* sebaiknya dikembangkan ke dalam bentuk aplikasi *mobile* berbasis android sehingga pasien dapat mengakses aplikasi pendaftaran dengan mudah dari *gadget* pasien. Kemudian saran kedua yaitu sebaiknya sistem rekam medis diintegrasikan

dengan sistem kasir sehingga setiap terapi dan jasa yang didapatkan pasien dapat langsung masuk ke tagihan pembayaran pada sistem kasir.



DAFTAR PUSTAKA

- Anon., 2018. *Guru 99*. [Online]
Available at: <https://www.guru99.com/compatibility-testing.html>
[Diakses 30 Juli 2018].
- Anon., 2018. *PowerMapper one click website tools*. [Online]
Available at: <https://www.powermapper.com/products/sortsite/>
[Diakses 30 Juli 2018].
- Arthur M. Langer, 2016. *Guide to Software Development Designing and Managing the Life Cycle*. 2nd ed. New York: Springer.
- Brooks, D. R., 2007. *An Introduction to HTML and Javascript for Scientist and Engineers*. London: Springer.
- Brooks, D. R., 2008. *Introduction to PHP for Scientists and Engineers Beyond Javascript*. London: Springer.
- Handayani, T. & Feoh, G., 2016. Perancangan Sistem Informasi Rekam Medis Berbasis Web (Studi Kasus Di Klinik Bersalin Sriati Kota Sungai Penuh – Jambi). pp. 226-336.
- Kementerian Kesehatan Republik Indonesia, 2008. *Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/MENKES/PER/III/2008*. [Online]
Available at: http://dinkes.surabaya.go.id/portal/files/permenkes/dok_dinkes_87.pdf
[Diakses 9 November 2018].
- Kahate, A., 2004. *Introduction to Database Management System*. India: Pearson.
- Khan, M. E., 2011. *Different Approaches to White Box Testing Technique for Finding Errors*. s.l.:Department of Information Technology Al Musanna College of Technology, Sultanate of Oman .
- Larman, C., 2001. *Applying UML and Patterns*. 2nd ed. s.l.:Prentice Hall.
- Madcoms, 2008. *PHP & MySQL untuk Pemula*. Yogyakarta: Andi Offset.
- Peicevic, A., 2017. *MySQL Introduction*. 1st ed. s.l.:Geek University Press.
- Pfaffenberger, B., Schafer, S. M., White, C. & Karow, B., 2004. *HTML, XHTML, and CSS Bible*. 3rd ed. Canada: Wiley Publishing, Inc..
- Pramesti, T. J. A., 2018. *Hukum Online*. [Online]
Available at: <http://www.hukumonline.com/klinik/detail/lt51fe16f7d4473/hak-pasien-dan-keluarganya-atas-rekam-medis>
[Diakses 30 Juli 2018].
- Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*,. 7th ed. New York: McGraw-Hill.

- Putri, N. S., Rochim, A. F. & Windasari, I. P., 2014. Perancangan Sistem Informasi Rekam Medis Rawat Inap Rumah Sakit Berbasis Web. pp. 1-13.
- Raymond McLeod, J. & Everett, G. D., 2007. *Software Testing Across the Entire Software Development Life Cycle*. New Jersey: IEEE Press.
- Roger S. Pressman, P. & Bruce R. Maxim, P., 2015. *Software Engineering A Practitioner's Approach*. 8th ed. New York: McGraw-Hill Education.
- Scott, T., Rundall, T. G., Vogt, T. M. & Hsu, J., 2007. *Implementing an Electronic Medical Record System successes, failure, lessons*. Seattle: Radcliffe Publishing.
- Sensuse, D. I., Kareen, P., Noprisson, H. & Pratama, M. O., 2017. *Success Factor for Health Information System Development*, pp. 162-167.
- Sommerville, I., 2011. *Software Engineering*. 9th ed. Boston: Pearson Education.
- Systems, S., 2007. *Using UML Part Two – Behavioral Modeling Diagrams*. [Online] Available at: <http://www.sparxsystems.com> [Diakses 5 September 2018].
- Winter, A. et al., 2011. *Health Information Systems Architectures and Strategies*. 2nd ed. London: Springer.

